NNT : 2023EMSEM032

THÈSE DE DOCTORAT

de Mines Saint-Etienne - Une école de l'IMT

Ecole Doctorale N°488

(Sciences, Ingénierie, Santé)

Spécialité de doctorat : Génie Industriel

Soutenue publiquement le 26 Septembre 2023 par :

Mahdi EL ALAOUI EL ABDELLAOUI

Methodology of Enterprise Modeling oriented Industry 4.0 : Application for Decision Support Systems in Manufacturing Systems

Méthodologie de Modélisation d'Entreprise orientée Industrie 4.0: Application aux Systèmes d'Aide à la Décision pour les Systèmes de Production

Devant un jury composé de :

DA CUNHA Catherine	*Professeure, Ecole Centrale de Nantes*	*Rapportrice*
SGARBOSSA Fabio	*Professeur, Université Norvégienne des Sciences et Technologies*	*Rapporteur*
VALLESPIR Bruno	*Professeur, Université de Bordeaux*	*Rapporteur*
DELORME Xavier	*Professeur, Mines Saint-Etienne*	*Directeur de thèse*
GRIMAUD Frédéric	*Professeur, Mines Saint-Etienne*	*Co-Directeur de thèse*
GIANESSI Paolo	*Maître Assistant, Mines Saint-Etienne*	*Co-encadrant de thèse*
STEPHENS Erwan	*Chef de projet, elm.leblanc-Groupe Bosch*	*Invité*

Affidavit

Je soussigné, Mahdi EL ALAOUI EL ABDELLAOUI, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Xavier DELORME et Frédéric GRIMAUD, dans le respect des principes d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect de la charte nationale de déontologie des métiers de la recherche.

Ce travail n'a pas été précédemment soumis dans sa globalité en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Saint-Etienne, le 22 juillet 2023

Ce travail de thèse est une œuvre de l'esprit, protégée par le droit d'auteur, tel que prévu aux articles L111-1 du CPI et suivants disposant que «L'auteur d'une œuvre de l'esprit jouit sur cette œuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous. [...]»Il est rappelé que par exception au droit d'auteur, la loi française autorise l'utilisation d'une œuvre divulguée, sans autorisateur de son auteur, suivant les conditions définies dans l'article L122-5 du CPI disposant que «Lorsque l'œuvre a été divulguée, l'auteur ne peut interdire [...] la représentation ou la reproduction d'extraits d'œuvres, [...] sous réserve que soient indiqués clairement le nom de l'auteur et la source [...] les analyses et courtes citations justifiées par le caractère critique, polémique, pédagogique, scientifique ou d'information de l'œuvre à laquelle elles sont incorporées [...] »

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski, Directeur de recherche	MATHEMATIQUES APPLIQUEES	M. Batton-Hubert, Professeure
MECANIQUE ET INGENIERIE	S. Drapier, Professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	P. Grosseau, Directeur de recherche	SCIENCES DES IMAGES ET DES FORMES	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	N. Absi, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	V. Laforest, Directrice de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'Etat ou d'une HDR)

ABSI	Nabil	DR	Génie industriel	CMP
AUGUSTO	Vincent	DR	Génie industriel	CIS
AVRIL	Stéphane	PR	Mécanique et ingénierie	CIS
BALBO	Flavien	PR	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR	Mathématiques appliquées	FAYOL
BEIGBEDER	Michel	MA	Informatique	FAYOL
BILAL	Blayac	DR	Sciences et génie de l'environnement	SPIN
BLAYAC	Sylvain	PR	Microélectronique	CMP
BOISSIER	Olivier	PR	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	DR	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR	Génie Industriel	FAYOL
BOUILLOT	Baptiste	MC	Sciences de l'ingénieur	SPIN
BRUCHON	Julien	PR	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	PR	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA	Génie industriel	Fayol
DELAFOSSE	David	PR	Sciences et génie des matériaux	SMS
DELORME	Xavier	PR	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	PR	Microélectronique	CMP
EL MRABET	Nadia	MA	Microélectronique	CMP
FAUCHEU	Jenny	MA	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	MR	Génie des Procédés	SPIN
FEILLET	Dominique	PR	Génie Industriel	CMP
FOREST	Valérie	PR	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GANSTER	Patrick	CR	Génie des Procédés	SMS
GAVET	Yann	MA	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA	Sciences et génie des matériaux	CIS
GONDRAN	Natacha	MA	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GRIMAUD	Frederic	EC	Génie mathématiques et industriel	FAYOL
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR	Génie des Procédés	SPIN
IANDOLA	Donata	MAA	Mécanique et ingénierie	CIS
ISMAILOVA	Esma	MC	Microélectronique	CMP
KERMOUCHE	Guillaume	PR	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	DR	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	DR	Mécanique et ingénierie	FAYOL
LIOTIER	Pierre-Jacques	MA	Mécanique et ingénierie	SMS
MEDINI	Khaled	EC	Sciences et génie de l'environnement	FAYOL
MOLIMARD	Jérôme	PR	Mécanique et ingénierie	CIS
MOULIN	Nicolas	MA	Mécanique et ingénierie	SMS
MOUTTE	Jacques	MR	Génie des Procédés	SPIN
NAVARRO	Laurent	MR	Mécanique et ingénierie	CIS
NEUBERT	Gilles	PR	Génie industriel	FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur-chercheur	Mécanique et ingénierie	CMP
O CONNOR	Rodney Philip	PR	Microélectronique	CMP
PEILLON	Sophie	MA	Sciences de gestion	FAYOL
PICARD	Gauthier	PR	Informatique	FAYOL
PINOLI	Jean Charles	PR	Sciences des Images et des Formes	SPIN
RIGAUD	Jean-Baptiste	MA	Microélectronique	CMP
POURCHEZ	Jérémy	DR	Génie des Procédés	CIS
ROUSSY	Agnès	MA	Microélectronique	CMP
SANAUR	Sébastien	MA	Microélectronique	CMP
SERRIS	Eric	IRD	Génie des Procédés	FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
VALDIVIESO	François	PR	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR	Génie industriel	CIS
YUGMA	Gallian	MR	Génie industriel	CMP
ZIMMERMAN	Antoine	EC	Informatique	FAYOL
RIEU	Mathilde	EC	Sciences et génie des matériaux	SPIN

Mise à jour : 04/02/2021

Methodology of enterprise modeling oriented Industry 4.0: Application for decision support systems in manufacturing systems

Author

Mahdi EL ALAOUI EL ABDELLAOUI

Supervisors

Xavier Delorme, Full Professor, Mines Saint-Étienne, France
Frédéric Grimaud, Full Professor, Mines Saint-Étienne, France
Paolo Gianessi, Associate Professor, Mines Saint-Étienne, France

Contact information

École des Mines de Saint-Étienne,
158 cours Fauriel, F-42023 Saint-Étienne cedex 2, France

Email address: webmaster@emse.fr
URL: https://www.mines-stetienne.fr/
Telephone: +33 (0)4 77 42 01 23
Fax: +33 (0)4 77 42 00 00

Copyright © 2023 by Mahdi EL ALAOUI EL ABDELLAOUI

Louange à Dieu, le Tout-Puissant,

Merci de m'avoir accordé la force, la patience et la clarté d'esprit nécessaires à l'accomplissement de ce travail. Que Sa guidance continue d'éclairer mon chemin.

À mes trés chérs parents,

Aucun mot ne saurait exprimer pleinement ma reconnaissance pour les sacrifices que vous avez consentis tout au long de mon parcours académique. Deux professeurs de mathématiques passionnés, vous avez été les premiers à éveiller en moi l'amour du savoir, et à tracer les premiers jalons de mon orientation scolaire et scientifique. Cette thèse n'aurait jamais vu le jour sans votre soutien indéfectible, moral, matériel et affectif. Que Dieu vous bénisse et protège, et vous accorde une longue vie pleine de santé et de sérénité. Je vous dédie ce travail, avec tout mon amour, en témoignage de ma profonde gratitude pour votre bienveillance et vos sacrifices.

À ma chère épouse,

Tu as été l'un des piliers majeurs de cette aventure. Ton soutien inconditionnel, tes encouragements quotidiens, ton assistance constante jusqu'aux dernières étapes ont été un moteur essentiel dans l'aboutissement de ce travail. Je te dédie cette thèse, au nom de notre amour et de notre complicité. Tu m'as toujours suivi dans mes idées, parfois un peu folles, et tu n'as jamais cessé de me pousser à aller plus loin, même dans les moments les plus difficiles. Merci de m'avoir permis de réaliser un rêve d'enfance. Tu es la seule qui a réellement compris pourquoi cette thèse était si importante pour moi. Ce travail t'est dédié, ma chérie, en témoignage de mon amour profond et de ma reconnaissance.

À ma fille, Eline, ma petite princesse,

Avant même ta naissance, tu étais déjà dans mes pensées à chaque page de cette thèse. J'ai écrit ces lignes en espérant qu'un jour, tu les liras avec fierté. Chaque mot, chaque phrase, chaque paragraphe a été rédigé avec amour et passion, en pensant à toi. Merci d'exister, d'avoir illuminé ma vie, même avant ton premier souffle. Je te dédie ce travail, ma princesse, comme un premier message d'inspiration et de confiance en toi. Que cette thèse t'accompagne, un jour, comme preuve que tout rêve mérite d'être poursuivi.

À mon cher frère,

Ton rôle a été capital durant ces années. Notre colocation à Lyon, les trajets matinaux vers Saint-Étienne, les discussions, les rires, ta présence, sans tout cela, les premières années de cette thèse auraient été bien plus dures. Je te remercie pour ton soutien discret mais essentiel, et pour ton humour, qui a souvent été ma meilleure motivation. Je te dédie cette thèse, en gage de reconnaissance et pour les moments partagés.

À Hajja Fatima,

Femme de coeur et d'esprit, tu inspires par ta sagesse et tes gestes quotidiens. Ton soutien indéfectible, ta bienveillance et ta motivation m'ont accompagné tout au long de cette aventure. Je te dédie cette thèse, avec estime et reconnaissance.

To God, the Almighty,

All praise be to Him for granting me the strength, patience, and clarity of mind to complete this journey. May His light continue to guide me and illuminate the path ahead.

To my beloved parents,

Words can scarcely capture the depth of my gratitude for the sacrifices you have made throughout my academic life. As mathematics teachers, you were the first to instill in me a love for knowledge and to lay the foundation of my scientific path. This thesis would not exist without your unwavering support—moral, material, and emotional. May God bless and protect you, and grant you long lives filled with health and serenity. With all my love, I dedicate this work to you, as a humble tribute to your tireless care and countless sacrifices.

To my loving and devoted wife,

You have been one of the greatest pillars of this journey. Your constant support, boundless encouragement, and daily presence were essential to the completion of this work. I dedicate this thesis to you, in the name of our love and the path we walk together. You embraced every one of my ideas, no matter how ambitious or eccentric, and always encouraged me to go further, even during the most trying times. Thank you for helping me fulfill a childhood dream. You are the only one who truly understood why this meant so much to me. This work is yours too, a symbol of my deep affection and gratitude.

To my daughter, Eline, my little princess,

Even before your first breath, you were present in every page of this thesis. I wrote each word with love, passion, and the hope that one day, you will read these lines with pride. Thank you for bringing light into my life before I ever held you in my arms. I dedicate this thesis to you as an early message of inspiration and belief in yourself. May this work serve as proof that every dream is worth pursuing.

To my dear brother,

From our shared flat in Lyon to the early morning commutes to Saint-Étienne, the laughter, the long conversations, and your quiet presence were essential during the early years of this thesis. Your support—often subtle yet so significant— and your sense of humor kept me going through difficult moments. I dedicate this thesis to you with heartfelt gratitude for everything you have done and the countless moments we have shared.

To Hajja Fatima,

A woman of heart and spirit, your wisdom have been a daily inspiration. Your unwavering support, kindness, and motivating words accompanied me every step of the way. I dedicate this work to you with deep respect and sincere thanks.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à mes encadrants de thèse. Ce travail n'aurait jamais vu le jour sans leur accompagnement précieux, leur rigueur scientifique et leur humanité.

Un grand merci à mon directeur de thèse, Professeur Xavier DELORME,

Le Big Boss. Que dire... Une personne brillante, pragmatique, dotée d'une rigueur intellectuelle impressionnante. Même en fin de thèse, tu arrivais encore à me surprendre par ta capacité à relier des sujets que je pensais disjoints.

Avec toi, tout est optimisé, même la présence : tu es là quand il faut, au bon moment. Tu sais pousser chacun dans ses retranchements, sans jamais briser, mais toujours en éveillant le meilleur. Tu es capable de cerner les compétences d'une personne en deux minutes. Tu sais exactement quand dire : *"Arrête de faire des lapalissades"* ou encore : *"Je sens que tu vas nous faire une Amine"* (Je te laisse, et toi seul, le privilège d'en comprendre le sens). Je me souviens de notre premier vol ensemble pour la conférence MIM à Berlin. Ta valise avait disparu, et n'est réapparue que plusieurs jours plus tard. Un début d'aventure à ton image : imprévisible, mais toujours sous contrôle.

Enfin, merci pour ton encadrement de qualité, ta bienveillance, ton humour subtil, et surtout ton soutien dans les moments les plus critiques de cette thèse. J'ai eu la chance de t'avoir comme directeur de thèse.

Un grand merci à mon co-directeur de thèse, Professeur Frédéric GRIMAUD,

Grâce à toi, de simples discussions de couloir pouvaient devenir des moments fondateurs. Une remarque, un mot, et me voilà à réfléchir pendant une semaine entière. Tu es le genre de personne qui pose un cadre initial —tel un entonnoir— puis laisse l'espace nécessaire à la réflexion, afin que les idées s'affinent et convergent d'elles-mêmes.

Je me souviendrai toujours de cette réunion en salle 524, après des semaines de recherche sur les décisions dans un MS. Tu avais dit : *"D'accord, c'est bien, c'est riche et complet... Et après ?"* Cette phrase, c'était un déclic. Tu m'as ramené à l'essentiel : structurer, relier, et modéliser. Après des mois de travail, les premières versions ont vu le jour. Et là, j'ai entendu : *"Là, on commence à avoir quelque chose qui a de la gueule"*. Tu m'as orienté vers GRAI, CIMOSA, et toutes ces approches : *"En gros, fais ton boulot."*

C'est à ce moment-là que j'ai compris que la modélisation d'enterprise allait devenir le coeur de ma thèse. J'ai plongé dans une nouvelle communauté, avec pour défi : faire progresser l'existant, dans un monde tourné vers l'industrie 4.0. Puis, en salle 514, lors d'une présentation, tu m'as lancé une phrase que je n'oublierai jamais : *"Ça, c'est une thèse."* Ce jour-là, j'ai compris que nous avions trouvé l'essence de mon sujet.

Tu m'as aussi enseigné l'humilité : *"Quoi que tu penses savoir, même après la thèse, tu ne seras jamais expert."* Une vérité que je ne peux que confirmer. Comme le dit un proverbe arabe : *"Plus on apprend, plus on découvre notre ignorance"*.

Enfin, je n'oublierai jamais ce moment de fierté lors de ta soutenance HDR, où tu as cité mon travail comme l'une des thèses dont tu étais le plus satisfait. Merci pour l'autonomie que tu m'as laissée, pour ta confiance, et pour tout le reste. J'espère sincèrement que nos chemins continueront de se croiser dans de futurs projets.

Un merci spécial à mon encadrant de thèse, Maître-Assistant Paolo GIANESSI,

Il m'est difficile de trouver les mots justes pour te remercier à la hauteur de ce que tu m'as apporté durant ces années de thèse. Je n'oublierai jamais les métaphores que tu inventais pour m'expliquer les concepts complexes — toujours en lien avec les chemins, les réseaux et l'optimisation... comme quoi le fameux VRP est toujours présent. Je me souviens encore des nuits blanches, des sacrifices, et de cette blague éternelle du *"Zuikov"* qui avait besoin d'un "∼\" pour respecter le nombre de pages... (Seuls toi et moi savons à quel point c'était sérieux—ou pas.)

J'ai énormément appris à tes côtés : me remettre en question, faire preuve de rigueur, comprendre que l'humilité est une force en science. Et moi qui croyais que la perfection était un ennemi... je t'ai trouvé encore plus exigeant que moi ! Nous pouvions passer 30 minutes à débattre d'un mot, d'une tournure de phrase, d'une idée à reformuler. Mais jamais je n'ai ressenti de hiérarchie pesante : tu as toujours su instaurer une relation de confiance, dans laquelle je ne me suis jamais senti "subordonné", mais plutôt accompagné par un mentor.

Même ta fameuse phrase : *"je te coupe les jambes"*, je la garde précieusement comme un clin d'oeil à notre complicité (et pour les services du travail : c'était évidemment une métaphore).

Ma reconnaissance va aux membres du jury et mes rapporteurs de thèse,

Je souhaite exprimer ma plus profonde gratitude aux membres du jury qui ont aimablement accepté d'évaluer ce travail : Professeure Catherine DA CUNHA, Professeur Fabio SGARBOSSA, et Professeur Bruno VALLéSPIR. Votre temps, vos remarques éclairées et vos retours précieux m'ont été d'une grande aide. Je vous en remercie sincèrement.

Mes remerciements chaleureux au groupe de recherche de Mines Saint-Étienne et aux équipes de l'entreprise elm.leblanc,

Je tiens à remercier tout particulièrement Emmanuel BRICARD, Emmanuel GRANGE, Stephens ERWAN, Hakim ZAHID, Christophe DE CARVALHO, Philippe LAFORGE, et Nicolas CASOETTO. Votre collaboration, vos conseils et votre soutien technique ont été essentiels tout au long de ce parcours.

Je remercie également Mireille BATTON-HUBERT, Audrey CERQUEUS et Damien LAMY pour leur soutien constant et leurs conseils avisés.

Une mention toute particulière au personnel administratif : Christine EXBRAYAT, Fanny AUBRUN, Joëlle GUELON, Amandine HIRONDEAU et Jean-FranÃ§ois TCHEBANOFF, dont l'aide précieuse et le dévouement ont grandement facilité de nombreuses démarches dans l'ombre.

Mes remerciements complices aux ami(e)s, et aux collègues,

Je tiens à adresser mes remerciements sincères à mes chers amis Mohammed-Amine ABDOUS et Sawssen SOUIDEN, pour leur présence indéfectible et leur soutien inestimable tout au long de cette aventure. Votre amitié a compté bien plus que je ne saurais le dire.

Également, je remercie chaleureusement mes collègues et camarades que j'ai eu le plaisir de côtoyer au fil de cette thèse, dans divers contextes et lieux : Yan, Nicolas N., Olivier, Vincent, Jean-Pierre, Adnane, Alessia, Nicolas J., Evariste, Léa, Sameh, Diego, Yannick, Amine, Stéphanie, et Niloofar. Votre présence, vos échanges, et les moments partagés ont enrichi ce parcours d'innombrables manières.

Mes remerciements du cœur aux membres de ma famille,

Enfin, je tiens à exprimer ma profonde gratitude pour le soutien indéfectible de ceux qui comptent le plus pour moi : ma mère et mon père, ma chère épouse et mon frère. Je ne peux imaginer ma vie sans votre amour. Je suis également profondément reconnaissant envers ma belle-mère, mes belles-soeurs, ainsi que mes oncles et tantes pour leur gentillesse et leur soutien.

Une mention toute spéciale revient à Salima pour son aide précieuse : tu as généreusement ouvert les portes de ton bureau pour moi, un espace où de nombreux paragraphes de cette thèse ont vu le jour et où j'ai préparé les dernières étapes de ma soutenance.

Je souhaite aussi remercier sincèrement Abdessadek, un ami cher qui a toujours été comme un frère pour moi, ainsi que mes cousins Wadie, Ahmed et Zakaria pour leur soutien constant. Je me considère véritablement chanceux d'être entouré d'une famille aussi aimante et solidaire.

Mahdi EL ALAOUI EL ABDELLAOUI

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my PhD supervisors. This thesis would never have come to life without their invaluable guidance, scientific rigor, and remarkable humanity.

Many thanks to my supervisor, Prof. Xavier DELORME,

The Big Boss. What can I say... A brilliant, pragmatic mind, with an impressive intellectual rigor. Even at the end of this PhD, you still managed to surprise me with your ability to connect topics I thought were worlds apart.

With you, everything is optimized—even your presence. You always show up at just the right moment. You know how to push people to their limits, not to break them, but to awaken their best selves. You have an uncanny talent for reading people in two minutes flat. And you always know the perfect time to say: "Stop stating the obvious," or "I can feel an Amine moment coming..." (I will let you—and only you—keep the privilege of understanding what that really means.) I remember our first flight together for the MIM conference in Berlin. Your suitcase was lost and was found a few days later. A start to our adventure that suits you perfectly: unpredictable, yet always under control.

Thank you for your high-quality mentorship, your kindness, your subtle humor, and above all, your support during the most critical moments of this journey. I was truly fortunate to have you as my advisor.

Many thanks to my supervisor, Prof. Frédéric GRIMAUD,

Thanks to you, even a casual hallway conversation could become a defining moment. A comment, a single word, and suddenly I'd find myself thinking about it for an entire week. You are the kind of person who provides an initial framework—like a funnel–and then creates the space for thoughtful exploration, allowing ideas to refine and converge gradually on their own.

I will never forget that meeting in Room 524, after weeks of researching decision-making in Manufacturing Systems. You said: "Okay, it is rich and comprehensive... but what comes next?" That sentence was a turning point. You brought me back to what matters: structure, coherence, and modeling. After months of work, the first drafts finally took shape. And then I heard: "Now we are starting to have something with real substance." You guided me toward GRAI, CIMOSA, and all

those foundational approaches: "Basically, just do your job." That was the moment I understood enterprise modeling would become the core of my thesis. I dived into a new community, with the challenge of creating something new from existing foundations, in a world moving toward Industry 4.0. Then one day, during a presentation in Room 514, you said something I will never forget: "Now this is a thesis." That day, I knew we had found the essence of my research.

You also taught me humility: "Whatever you think you know—even after the PhD—you'll never truly be an expert." A truth I can only confirm. As an old Arabic proverb says: "The more we learn, the more we discover our ignorance."

Finally, I will always remember the pride I felt during your HDR defense, when you mentioned my thesis as one of the works you were most proud of. Thank you for the autonomy you gave me, for your trust, and for everything else. I truly hope our paths will cross again in future projects.

Special thanks to my supervisor, Assoc.Prof Paolo GIANESSI,

It is hard to find the right words to thank you for everything you brought into my life during these five years of research. I will never forget the metaphors you invented to explain complex concepts—always involving paths, sources, and optimization... like that famous VRP that always returns to its origin. I still remember the sleepless nights, the sacrifices, and the eternal joke about "Zuikov" needing a "$\sim\backslash$" just to meet the page count... (Only you and I know how serious—or not—that really was.)

I learned so much by your side: how to question myself, how to work with rigor, how to understand that humility is a strength in science. And just when I thought perfectionism was my weakness... I found in you someone even more demanding than myself! We could spend 30 minutes discussing a single word, a turn of phrase, or a reformulated idea. Yet never once did I feel the weight of hierarchy—you always fostered a relationship built on trust, where I never felt like a "subordinate", but rather a mentee guided by a mentor.

Even your famous line: "I am cutting your legs off," remains with me as a symbol of our complicity. (And for HR purposes: yes, it was a metaphor.)

My acknowledgment goes to the jury members and my thesis reviewers,

I would like to express my deepest gratitude to the members of the jury who kindly agreed to evaluate this work: Prof. Catherine DA CUNHA, Prof. Fabio SGARBOSSA, and Prof. Bruno VALLÉSPIR. Your time, insightful comments, and valuable feedback are sincerely appreciated.

My heartfelt thanks go to the research group at Mines Saint-Étienne and the elm.leblanc company teams,

With special appreciation to Emmanuel BRICARD, Emmanuel GRANGE, Stephens ERWAN, Hakim ZAHID, Christophe DE CARVALHO, Philippe LAFORGE, and Nicolas

CASOETTO. Your collaboration, guidance, and technical support have been essential throughout this journey.

I also warmly thank Mireille BATTON-HUBERT, Audrey CERQUEUS, and Damien LAMY for their continuous support and thoughtful advice. A very special acknowledgment goes to the administrative staff: Christine EXBRAYAT, Fanny AUBRUN, Joëlle GUELON, Amandine HIRONDEAU et Jean-FranÃ§ois TCHEBANOFF, whose help and dedication made many things easier behind the scenes.

My warm and heartfelt thanks to my friends and colleagues,

I sincerely thank my dear friends Mohammed-Amine Abdous and Sawssen Souiden for always being there in so many helpful ways. Your friendship and encouragement meant more than I can say.

I would also like to thank my colleagues and peers I had the pleasure of meeting throughout this thesis in various contexts and places: Yan, Nicolas N., Olivier, Vincent, Jean-Pierre, Adnane, Alessia, Nicolas J., Evariste, Léa, Sameh, Diego, Yannick, Amine, Stéphanie, et Niloofar. Your presence, collaboration, and shared experiences enriched this journey in countless ways.

My deepest gratitude to the members of my family,

Finally, I would like to express my heartfelt appreciation for the unwavering support of those who mean the world to me: my mother and father, my beloved wife, and my brother. I cannot imagine life without your love. I am also deeply grateful to my mother-in-law, my sisters-in-law, as well as my uncles and aunts for their kindness and support.

A very special thanks goes to Salima for her invaluable help—you generously opened the doors of your office to me, providing a space where many paragraphs of this thesis were written and where I prepared the final stages of my defense.

I would also like to sincerely thank Abdessadek, a dear friend who has always felt like a brother, and my cousins Wadie, Ahmed, and Zakaria for their continued encouragement. I am truly lucky to be surrounded by such a supportive and loving family.

Mahdi EL ALAOUI EL ABDELLAOUI

Table des Matières

Liste des Figures **xl**

Liste des Tables **xlviii**

Liste des Algorithmes **liii**

Introduction (version française) **1**

Introduction **7**

1 Systèmes d'Aide à la Décision dans les Systèmes de Production **11**

1.1 Introduction 12

1.2 Systèmes de Production: de la Production à l'Information 14

1.2.1 Classification des Systèmes de Production 15

1.2.1.1 Types des Systèmes de Production 15

1.2.1.2 Comparaison des Types des Systèmes de Production . . . 16

1.2.2 Défis des Systèmes de Production en Industrie 4.0 17

1.2.3 Émergence des Nouvelles Technologies dans les Systèmes de Production 20

1.2.3.1 Technologies Logicielles 21

1.2.3.2 Technologies Matérielles 23

1.2.3.3 Nouvelles Technologies pour la Digitalisation de la Production 24

1.2.3.4 Comparaison d'un Système Cyber-Physique et d'un Jumeau Numérique 27

1.2.3.5 Conclusion sur les Impacts des Nouvelles Technologies . . 27

1.3 Cadres de Modélisation d'Entreprise 28

1.3.1 Concepts et Terminologie de Modélisation d'Entreprise 28

1.3.2 Le Rôle de Modélisation d'Entreprise 32

1.3.3 Cadres Classiques de Modélisation d'Entreprise 32

1.3.4 Nouveaux Cadres de Modélisation d'Entreprise en Industrie 4.0 . . 34

1.3.4.1 Architecture de Référence pour Modélisation Industrie 4.0 34

1.3.4.2 Architecture de Référence pour l'Internet Industrielle . . . 35
1.3.4.3 Comparaison des Cadres Existants de Modélisation d'Entreprise . . . 35
1.3.5 Méthodes et Outils de Modélisation d'Entreprise . . . 36
1.3.5.1 Méthodes de Modélisation . . . 36
1.3.5.2 Outils de Modélisation . . . 38
1.4 Prise de Décision et Aide à Décision dans les Systèmes de Production . . . 42
1.4.1 Importance des Systèmes d'Aide à la Décision dans les Systèmes de Production . . . 44
1.4.2 Types des Systèmes d'Aide à la Décision . . . 44
1.4.3 Modélisation Quantitative . . . 45
1.4.3.1 Modèles de Performance Analytiques . . . 45
1.4.3.2 Modèles d'Optimisation . . . 46
1.4.3.3 Modèles de Simulation . . . 46
1.4.4 Prise de Décision et Systèmes d'Aide à la Décision dans l'Industrie 4.0 . . . 49
1.5 Questions de Recherches Ouvertes . . . 50

2 Problèmes de Décision dans les Systèmes de Production et leurs Évolutions dans l'Industrie 4.0 57
2.1 Introduction . . . 58
2.2 Problèmes de Décision dans les Systèmes de Production . . . 59
2.2.1 Décisions liées à l'Activité Production . . . 59
2.2.1.1 Stratégie de Production . . . 59
2.2.1.2 Planification des Capacités . . . 60
2.2.1.3 Configuration de Ligne . . . 60
2.2.1.4 Conception de la Disposition . . . 61
2.2.1.5 Reconfiguration de Ligne . . . 61
2.2.1.6 Dimensionnement du Lot . . . 62
2.2.1.7 Planification et Ordonnancement . . . 62
2.2.1.8 Maintenance et Contrôle de Processus . . . 63
2.2.2 Décisions liées à l'Activité Approvisionnement . . . 64
2.2.2.1 Configuration du Système d'Alimentation . . . 64
2.2.2.2 Politique d'Alimentation de Ligne . . . 65
2.2.2.3 Stock . . . 65
2.2.2.4 Chargement . . . 66
2.2.3 Décisions liées à l'Activité Stockage . . . 66
2.2.3.1 Dimensionnement et Taille de l'entrepôt . . . 67
2.2.3.2 Sélection d'Equipement . . . 67
2.2.3.3 Disposition du Département . . . 68

2.2.3.4 Stratégie Opérationnelle 69
2.2.3.5 Problème d'Affectation d'Emplacement de Stockage 69
2.2.3.6 Zonage 70
2.2.3.7 Préparation de Commandes 70
2.3 Transformation de l'Ancien Scénario Industriel vers le Scénario Industrie 4.0 71
2.3.1 Évolution Technique 74
2.3.1.1 Standardisation de l'Échange des Données 74
2.3.1.2 Nouvelles Techniques et Algorithmes 76
2.3.2 Évolution Organisationnelle 77
2.3.2.1 Stratégie Leagile 78
2.3.2.2 Système de Production Centré sur l'Humain 79
2.3.2.3 Système de Production Adaptable 80
2.3.3 Évolution Transversale 81
2.3.3.1 Surveillance en Temps Réel 81
2.3.3.2 Approche de Contrôle Décentralisé 82
2.3.3.3 Concept d'Indicateur Clé de Performance 83
2.3.3.4 Indicateurs Clés Classiques de Performance dans les Systèmes de Production 84
2.3.3.5 Indicateurs Clés de Performance Orientés Durabilité et Agilité dans les Systèmes de Production 84
2.4 Évolution des Probèmes de Décision dans les Systèmes de Production en Industrie 4.0 87
2.4.1 Évolution de l'Horizon Temporel des Probèmes de Décision dans les Systèmes de Production 87
2.4.1.1 L'Horizon Temps Réel 88
2.4.1.2 Évolution de l'Horizon Temporel de Quelques Décision . . 89
2.4.2 Évolution du Type de Systèmes de Production 93
2.5 Conclusion 93

3 Modélisation des Problèmes de Décision dans les Systèmes de Production et leurs Intéractions 95
3.1 Introduction 96
3.2 Une Proçédure à Trois Étapes pour Étudier la Prise de Décision dans les Systèmes de Production 98
3.2.1 Concepts de Base 98
3.2.2 Modèle Conceptuel de la Vue Fonction d'un Système de Production 101
3.2.3 Introduction d'une Proçédure à Trois Étapes pour Étudier les Décisions dans les Systèmes de Production 102
3.3 Identification et Classification de Décision 102
3.4 Définition de Décision 103

3.4.1 Modèle Conceptuel de la Vue Décision d'un Système de Production 103
3.4.2 Modèle Conceptuel de la Vue Information d'un Système de Production 105
3.4.3 Modèle Conceptuel de la Vue Performance d'un Système de Production . . . 106
3.4.4 Propriètés des Problèmes de Décision . . . 108
3.5 Interactions de Décision . . . 108
3.5.2 Types d'Interactions de Décision . . . 113
3.5.3 Règles pour Étudier les Interactions de Décision . . . 114
3.5.3.1 Règles Génériques pour Étudier les Interactions de Décision 116
3.5.3.2 Règles Spécifiques pour Étudier les Interactions de Décision 116
3.5.4 Interactions de Décision Dépendantes du Type de Système de Production . . . 117
3.5.4.1 Interaction entre l'Équilibrage de Ligne et la Planification et Ordonnancement . . . 117
3.5.4.2 Intéraction entre la Planification des Capacités et l'Équilibrage/Réquilibrage de Ligne . . . 118
3.6 Un Outil Graphique pour Représenter les Décisions dans les Systèmes de Production et Comment ils Interagissent entre Elles . . . 119
3.7 Utilisation de l'Outil Cartographie pour Représenter l'Évolution des Problèmes de Décisions et les Interactions entre Eux dans le Contexte Industrie 4.0 . . . 121
3.7.1 Évolution dans l'Horizon Temporel des de Décisions . . . 121
3.7.2 Évolution des Interactions de Décision . . . 133
3.8 Conclusion . . . 136

4 Nouvelle Méthodologie de Modélisation d'Enteprise Orientée Industrie 4.0 139
4.1 Introduction . . . 140
4.2 Nouvelles Approches de Modélisation pour la Modélisation d'Enteprise en Industrie 4.0 . . . 142
4.3 Méthodologie de Modélisation d'Enteprise Orientée Industrie 4.0 (MEMO I4.0) . . . 143
4.3.1 Principes Clés de MEMO I4.0 . . . 144
4.3.1.1 Principe de Modularité . . . 144
4.3.1.2 Principe d'Agilité . . . 145
4.3.1.3 Principe de Robustesse . . . 145
4.3.1.4 Principe d'Intéropérabilité . . . 145
4.3.2 Objectives de Modélisation de MEMO I4.0 . . . 145
4.3.2.1 Processus de Modélisation . . . 146
4.3.2.2 Niveau de Modélisation . . . 146

4.3.2.3 Granularité de Modélisation . . . 146
4.3.2.4 Cohérence en Modélisation . . . 147
4.3.3.0 Dimensions de MEMO I4.0 . . . 148
4.3.3.1 Dimension Cycle de Vie . . . 148
4.3.3.2 Dimension Généricité . . . 149
4.3.3.3 Dimension Échelle . . . 149
4.3.3.4 Dimension Vue . . . 149
4.3.4 Spécifications et Objectifs de MEMO I4.0 . . . 151
4.4 Application de MEMO I4.0: Étape Stratégique/Tactique . . . 153
4.4.1 Phase Définition: Exigences et Spécifications de la Modélisation . . 155
4.4.1.1 Définition des Objectifs de Modélisation et du Périmètre . 155
4.4.1.2 Description de la Vision d'Entreprise, de ses Lignes directrices et son ADN . . . 157
4.4.1.3 Décomposition en Modules et Définition de leurs Description Fonctionnelle et Hypothèses . . . 158
4.4.1.4 Séparation des Flux et Définition des Sous-modules . . . 159
4.4.1.5 Construction des Sénarios de Modélisation . . . 159
4.4.1.6 Spécification du Contexte de Modélisation et du Degré de Généricité . . . 160
4.4.1.7 Identification des Décisions liées aux Modules . . . 161
4.4.1.8 Définition des Indicateurs Clés de Performance du Module 161
4.4.1.9 Spécification du Niveau d'Échelle . . . 161
4.4.1.10 Définition des Départements liés aux Modules . . . 162
4.4.1.11 Définition et Association des Informations des Modules avec les Caractéristiques du Système de Production . . . 162
4.4.1.12 Identification des Données Brutes et leur Organisation par Département . . . 163
4.4.1.13 Identification des Élements Impliqués dans Chaque Module 164
4.4.2 Phase Conception: Instanciation des Métamodèles des Vues . . . 165
4.4.2.1 Décision des Choix de Modélisation de Chaque Sous-module 165
4.4.2.2 Choix de l'Approche de Modélisation de Chaque Sous-module . . . 165
4.4.2.3 Spécification des Évènements déclencheurs de la Fonction 166
4.4.2.4 Décisions Impactées par les Évènements de Mise á Jour . . 166
4.4.2.5 Spécification des Évènements Impactant sur Chaque Information . . . 167
4.4.2.6 Construction de la Structure et Format des Indicateurs Clés de Performance . . . 167
4.4.2.7 Les Structures des Données . . . 167

4.4.2.8 Dessiner un Organigramme . . . 168
4.4.2.9 Instanciation des Métamodèles des Vues . . . 169
4.4.3 Phase Implémentation: Vérification et Validation du Comportement du Modéle de Chaque Module . . . 169
4.4.3.1 Choix de la Solution Technique pour Capturer les Données Brutes . . . 170
4.4.3.2 Création en Bases de Données des Tables Requises par le Modèle de Données . . . 170
4.4.3.3 Création de la Couche d'Accès aux Données . . . 171
4.4.3.4 Construction de la Couche du Modèle Métier . . . 171
4.4.3.5 Création de la Couche Correspondance . . . 171
4.4.3.6 Écriture du Pseudo-Code des Modules . . . 172
4.4.3.7 Construction d'un Modèle d'Optimisation ou d'un Arbre de Décision . . . 172
4.4.3.8 Choix de l'Approche de Résolution . . . 173
4.4.3.9 Établissement des Expérimentations Numériques . . . 174
4.4.3.10 Construction d'un Modèle de Simulation des Modules . . . 174
4.4.3.11 Correspondance des Variables de Décision du Modèle d'Optimisation et des Attributs du Modèle de Simulation . 175
4.4.3.12 Création des Fichiers de Traçabilité . . . 175
4.4.3.13 Lancement des Expérimentations . . . 176
4.4.3.14 Exécution des Tests extrêmes . . . 176
4.4.3.15 Affichage des Indicateurs Clés de Performance . . . 176
4.4.3.16 Vérification de l'Agilité et la Flexibilité du Module . . . 176
4.4.4 Phase Maintenance: Versionnage du Modèle de Chaque Module et Génération de la Documentation Correspondante . . . 177
4.4.4.1 Documentation et Versionnage des Modèles du Module . . 177
4.4.4.2 Vérification de la Compatibilité entre les Modéles de Vues des Modules . . . 177
4.4.4.3 Organisation des Réunions de Revue avec l'Équipe Projet 177
4.4.4.4 Mise à Jour de la Documentation et Versionnage des Modèles des Modules . . . 178
4.5 Application de MEMO I4.0: Étape Opérationnelle . . . 178
4.5.1 Phase Définition . . . 178
4.5.2 Phase Conception . . . 178
4.5.3 Phase Implémentation . . . 179
4.5.4 Phase Maintenance . . . 180
4.6 Conclusion . . . 180

5 Application de MEMO I4.0 dans un Environnement Industriel Réel: une Étude de Cas 183

5.1 Introduction 184

5.2 Étape Stratégique/Tactique: Développement des Modules 185

5.2.1 Phase Définition: Exigences et Spécifications de Modélisation . . . 187

5.2.1.1 Définition des Objectifs de Modélisation et du Périmètre . 187

5.2.1.2 Description de la Vision d'Entreprise, de ses Lignes directrices et son ADN 187

5.2.1.3 Décomposition en Modules et Définition de leurs Description Fonctionnelle et Hypothèses 189

5.2.1.4 Séparation des Flux et Définition des Sous-modules 191

5.2.1.5 Construction des Sénarios de Modélisation 191

5.2.1.6 Spécification du Contexte de Modélisation et du Degré de Généricité 193

5.2.1.7 Identification des Décisions liées aux Modules 193

5.2.1.8 Définition des Indicateurs Clés de Performance du Module 194

5.2.1.9 Spécification du Niveau d'Échelle 194

5.2.1.10 Définition des Départements liés aux Modules 195

5.2.1.11 Définition et Association des Informations des Modules avec les Caractéristiques du Système de Production 195

5.2.1.12 Identification des Données Brutes et leur Organisation par Département 196

5.2.1.13 Identification des Élements Impliqués dans Chaque Module 196

5.2.2 Phase Conception: Instanciation des Métamodèles des Vues 197

5.2.2.1 Décision des Choix de Modélisation de Chaque Sous-module 197

5.2.2.2 Choix de l'Approche de Modélisation de Chaque Sous-module 198

5.2.2.3 Spécification des Évènements déclencheurs de la Fonction 199

5.2.2.4 Décisions Impactées par les Évènements de Mise á Jour . . 200

5.2.2.5 Spécification des Évènements Impactant sur Chaque Information 200

5.2.2.6 Construction de la Structure et Format des Indicateurs Clés de Performance 201

5.2.2.7 Les Structures des Données 201

5.2.2.8 Dessiner un Organigramme 201

5.2.2.9 Instanciation des Métamodèles des Vues 202

5.2.3 Phase Implémentation: Vérification et Validation du Comportement du Modéle de Chaque Module 205

5.2.3.1 Choix de la Solution Technique pour Capturer les Données Brutes . . . 206
5.2.3.2 Création en Bases de Données des Tables Requises par le Modèle de Données . . . 206
5.2.3.3 Création de la Couche d'Accès aux Données . . . 206
5.2.3.4 Construction de la Couche du Modèle Métier . . . 207
5.2.3.5 Création de la Couche Correspondance . . . 208
5.2.3.6 Écriture du Pseudo-Code des Modules . . . 208
5.2.3.7 Construction et Résolution d'un Modèle d'Optimisation et Établissement des Expérimentations Numériques . . . 209
5.2.3.8 Construction d'un Modèle de Simulation des Modules . . . 209
5.2.3.9 Création des Fichiers de Traçabilité . . . 212
5.2.3.10 Lancement des Expérimentations . . . 212
5.2.3.11 Exécution des Tests extrêmes . . . 213
5.2.3.12 Affichage des Indicateurs Clés de Performance . . . 214
5.2.3.13 Vérification de l'Agilité et la Flexibilité du Module . . . 214
5.2.4 Phase Maintenance: Versionnage du Modèle de Chaque Module et Génération de la Documentation Correspondante . . . 215
5.2.4.1 Documentation et Versionnage des Modèles du Module . . . 215
5.2.4.2 Vérification de la Compatibilité entre les Modéles de Vues des Modules . . . 215
5.2.4.3 Organisation des Réunions de Revue avec l'Équipe Projet 216
5.2.4.4 Mise à Jour de la Documentation et Versionnage des Modèles des Modules . . . 216
5.3 Étape Opérationnelle: Investigation de la Logistique Interne par Simulation 216
5.3.1 Phase Définition . . . 217
5.3.2 Phase Conception . . . 217
5.3.3 Phase Implémentation . . . 218
5.3.4 Phase Maintenance . . . 219
5.4 Conclusion . . . 220

Conclusion & Perspectives (version française) 223

Conclusion & Perspectives 231

Références 237

A Publications 275

B MS Features 277

C Conceptual Models **291**
C.1 Modèles Conceptuels des Vues: Métamodèles . . . 291
C.1.1 Modèle Conceptuel de l'Intégration Partielle des Vues d'un Système de Production . . . 291
C.1.2 Modèle Conceptuel de la Vue Processus d'un Système de Production 292
C.1.3 Modèle Conceptuel de la Vue Données d'un Système de Production 293
C.1.4 Modèle Conceptuel de la Vue Ressource & Organisation d'un Système de Production . . . 295
C.1.5 Modèle Conceptuel de l'Intégration de Toutes les Vues d'un Système de Production . . . 297
C.2 Modèles Conceptuels des Principes de MEMO I4.0 . . . 299

D Exemples d'Approches de Modélisation **303**
D.1 Approche Indépendante: Exemple de Modélisation de la Politique d'Alimentation de Ligne . . . 303
D.2 Approche Intérmédiaire: Exemple de Modélisation de la Stratégie de Stockage 304
D.3 Approche Intégrée: Exemple de Modélisation de la Politique de Production 305

E Application de MEMO I4.0: Exemple Didactique de l'Entreprise E1 **307**
E.1 Feuille de Route pour l'Implémentation de MEMO I4.0 (version détaillée) . 307
E.2 Phase Définition de MEMO I4.0: Exemple Didactique de l'Entreprise E1 . 309
E.2.1 Hypothèses liées aux Modules de l'activité Production de E1 309
E.2.2 Choix du Niveau d'Échelle du Module d'Assemblage de E1 309
E.2.3 Approche de Modélisation du Sous-module Physique du Module d'Assemblage de E1 . . . 311
E.3 Phase Conception de MEMO I4.0: Exemple Didactique de E1 313
E.3.1 Les Choix de Modélisation de Chaque Sous-module des Modules de E1 . . . 313
E.3.2 Instanciation du Métamodèle de la Vue Fonction de E1 315
E.4 Phase Implémentation de MEMO I4.0: Exemple Didactique de E1 316
E.4.1 Correspondance des Variables de Décision du Module Planifier de E1 et les Attributs du Modèle de Simulation . . . 316
E.4.2 Liste des Notations Utilisées dans les Pseudo-Codes 318
E.4.3 Liste des Algorithmes (les Pseudo-Codes) . . . 319

F Application de MEMO I4.0: Entreprise elm.leblanc **325**
F.1 Phase Définition . . . 325
F.1.1 Liste des Scénarios de Modélisation des Modules du Projet MEMO I4.0 au Sein de l'Entreprise elm.leblanc . . . 325
F.1.2 Liste des Tables liées aux Données de Fonction des Modules 333

F.1.3 Identification des Données Brutes et leur Organisation par Département . . . 338
F.2 Phase Conception . . . 338
F.3 Phase Implémentation . . . 345
F.3.1 Création de la Couche d'Accès aux Données . . . 345
F.3.2 Liste des Algorithmes (les Pseudo-Codes) . . . 346
F.3.3 Construction d'un Modèle de Simulation des Modules: Résultats . . . 350
F.3.4 Analyse de Sensibilité pour les Variables: X1 et X2 . . . 352
F.4 Phase Maintenance . . . 354
F.4.1 Documentation et Versionnage des Modèles du Module . . . 354
F.5 Étape Opérationnelle: Analyse par Simulation de l'Augmentation de la Demande Client . . . 358
F.5.1 Phase Définition . . . 358
F.5.2 Phase Conception . . . 358
F.5.3 Phase Implémentation . . . 359
F.5.4 Phase Maintenance . . . 359
F.6.0 Étape Opérationnelle: Couplage de la Simulation et de l'Optimisation pour Traiter le Problème d'Équilibrage de Ligne avec des Indicateurs Clés de Performances 4.0 (Ergonomie) . . . 360
F.6.1 Phase Définition . . . 360
F.6.2 Retour au Cycle de Développement . . . 360
F.6.2.1 Phase Définition de l'Étape Stratégique/Tactique . . . 361
F.6.2.2 Phase Conception de l'Étape Stratégique/Tactique . . . 361
F.6.2.3 Phase Implémentation de l'Étape Stratégique/Tactique . . . 362
F.6.2.4 Phase Maintenance de l'Étape Stratégique/Tactique . . . 367
F.6.3 Phase Conception . . . 367
F.6.4 Phase Implémentation . . . 367
F.6.5 Phase Maintenance . . . 369
F.7 Instanciation des Modèles . . . 370

G Liste des Utilitaires **377**

Résumé **395**

Contents

List of Figures **xli**

List of Tables **xlix**

List of Algorithms **lv**

Introduction (french version) **1**

Introduction **7**

1 Decision Support Systems in Manufacturing Systems **11**

1.1 Introduction 12

1.2 Manufacturing Systems: from Production to Information 14

1.2.1 Classification of Manufacturing Systems 15

1.2.1.1 Manufacturing Systems Types 15

1.2.1.2 Comparison of Manufacturing Systems Types 16

1.2.2 Challenges of Manufacturing Systems in the Industry 4.0 17

1.2.3 Emergence of New Technologies in Manufacturing Systems 20

1.2.3.1 Software Technologies 21

1.2.3.2 Hardware Technologies 23

1.2.3.3 New Technologies for Computerization of Manufacturing . 24

1.2.3.4 Comparison of Cyber-Physical System and Digital Twin . 27

1.2.3.5 Conclusion on the Impact of New Technologies 27

1.3 Enterprise Modeling Frameworks 28

1.3.1 Concepts and Terminology of Enterprise Modeling 28

1.3.2 The Role of Enterprise Modeling 32

1.3.3 Classical Frameworks of Enterprise Modeling 32

1.3.3.1 GRAI Integrated Methodology 33

1.3.3.2 Computer Integrated Manufacturing Open System Architecture 33

1.3.3.3 Generalised Enterprise Reference Architecture and Methodology 33

1.3.4 New Frameworks of Enterprise Modeling in Industry 4.0 34
1.3.4.1 Reference Architecture Modeling Industry 4.0 34
1.3.4.2 Industrial Internet Reference Architecture 35
1.3.4.3 Comparison of Existing Entreprise Modeling Frameworks . 35
1.3.5 Methods and Tools of Enterprise Modeling 36
1.3.5.1 Modeling Methods 36
1.3.5.2 Modeling Tools 38
1.4 Decision Making and Decision-Support in Manufacturing Systems 42
1.4.1 Importance of Decision-Support Systems in Manufacturing Systems 44
1.4.2 Types of Decision Support Systems 44
1.4.3 Quantitative Modeling 45
1.4.3.1 Analytical Performance Models 45
1.4.3.2 Optimization Models 46
1.4.3.3 Simulation Models 46
1.4.4 Decision Making and Decision Support Systems in the Industry 4.0 49
1.5 Open Research Questions 50

2 Decision Problems in Manufacturing Systems and their Evolutions in the Industry 4.0 57
2.1 Introduction 58
2.2 Decision Problems in Manufacturing Systems 59
2.2.1 Decisions related to Production Activity 59
2.2.1.1 Production Strategy 59
2.2.1.2 Capacity Planning 60
2.2.1.3 Line Configuration 60
2.2.1.4 Layout Design 61
2.2.1.5 Line Reconfiguration 61
2.2.1.6 Lot Sizing 62
2.2.1.7 Sequencing and Scheduling 62
2.2.1.8 Maintenance and Process Control 63
2.2.2 Decisions related to the Supply Activity 64
2.2.2.1 Feeding System Configuration 64
2.2.2.2 Line Feeding Policy 65
2.2.2.3 Inventory 65
2.2.2.4 Loading 66
2.2.3 Decisions related to the Storage Activity 66
2.2.3.1 Warehouse Dimensioning and Sizing 67
2.2.3.2 Equipment Selection 67
2.2.3.3 Department Layout 68
2.2.3.4 Operation Strategy 69

2.2.3.5 Storage Location Assignment Problem (SLAP) 69
2.2.3.6 Zoning 70
2.2.3.7 Order Picking 70
2.3 Transformation of the Old Industrial Scenario into Industry 4.0 Scenario . 71
2.3.1 Technical Evolution 74
2.3.1.1 Standardization of Data Exchange 74
2.3.1.2 New Techniques and Algorithms 76
2.3.2 Organizational Evolution 77
2.3.2.1 Leagile Strategy 78
2.3.2.2 Human-Centered MS 79
2.3.2.3 Adaptable MS 80
2.3.3 Transversal Evolution 81
2.3.3.1 Real-Time Monitoring 81
2.3.3.2 Decentralized Control Approach 82
2.3.3.3 Concept of KPI 83
2.3.3.4 Classical KPIs in MSs 84
2.3.3.5 Sustainability-Oriented KPIs and Agility in MSs 84
2.4 Evolution of Decision Problems in Manufacturing Systems in Industry 4.0 . 87
2.4.1 Evolution of Time Horizon of Decision Problems in MS 87
2.4.1.1 Real-Time Horizon 88
2.4.1.2 Evolution of Time Horizon of Some Decisions 89
2.4.2 Evolution of Manufacturing Systems Type 93
2.5 Conclusion 93

3 Modeling of Decision Problems in Manufacturing Systems and their Interactions 95
3.1 Introduction 96
3.2 A Three-Step Procedure to Study Decision Making in Manufacturing Systems 98
3.2.1 Basic Concepts 98
3.2.2 Conceptual Model of the Function View of an MS 101
3.2.3 Introduction of a Three-Step Procedure to Study Decisions in MSs . 102
3.3 Decision Identification and Classification 102
3.4 Decision Definition 103
3.4.1 Conceptual Model of Decision View of an MS 103
3.4.2 Conceptual Model of Information View of an MS 105
3.4.3 Conceptual Model of Performance View of an MS 106
3.4.4 Properties of Decision Problems 108
3.5 Decision Interactions 108
3.5.1 Some Decision Interactions Linked to Plant Activities 110
3.5.1.1 Some Interactions of Production Activity Decisions 110

3.5.1.2 Some Interactions among Supply Activity Decisions 112
3.5.1.3 Some Interactions among Storage Activity Decisions . . . 113
3.5.2 Types of Decision Interactions 113
3.5.3 Rules to Study Decision Interactions 114
3.5.3.1 Generic Rules to study decision interactions 116
3.5.3.2 Specific Rules to Study Decision Interactions 116
3.5.4 MS Type-Dependent Decision Interactions 117
3.5.4.1 Interaction between the Line Balancing and the Sequencing & Scheduling 117
3.5.4.2 Interaction between the Capacity Planning and the Line Balancing/Rebalancing 118
3.6 A Graphical Tool to Represent Decisions in MSs and How they Interact with Each Other 119
3.7 Using the Cartography Tool to Represent the Evolution of Decision Problems and Interactions among them in the Industry 4.0 Context 121
3.7.1 Evolution in the Time Horizon of Decisions 121
3.7.2 Evolution of Decision Interactions 133
3.8 Conclusion 136

4 New Methodology of Enterprise Modeling Oriented Industry 4.0 139
4.1 Introduction 140
4.2 New Modeling Approaches for Enterprise Modeling in Industry 4.0 142
4.3 Methodology of Enterprise Modeling Oriented I4.0 (MEMO I4.0) 143
4.3.1 Key Principles of MEMO I4.0 144
4.3.1.1 Modularity Principle 144
4.3.1.2 Agility Principle 145
4.3.1.3 Robustness Principle 145
4.3.1.4 Interoperability Principle 145
4.3.2 Modeling Objectives of MEMO I4.0 145
4.3.2.1 Modeling Process 146
4.3.2.2 Modeling Level 146
4.3.2.3 Modeling Granularity 146
4.3.2.4 Modeling Consistency 147
4.3.3 Dimensions of MEMO I4.0 148
4.3.3.1 Life Cycle Dimension 148
4.3.3.2 Genericity Dimension 149
4.3.3.3 Scale Dimension 149
4.3.3.4 View Dimension 149
4.3.4 Specifications and Purpose of MEMO I4.0 151
4.4 Application of MEMO I4.0: Strategical/Tactical Stage 153

4.4.1 Definition Phase: Requirements and Specifications of Modeling . . . 155
4.4.1.1 Definition of Modeling Objectives and Scope 155
4.4.1.2 Description of the Enterprise Vision, Guidelines & DNA . 157
4.4.1.3 Decomposition into Modules and Definition of their Functional Description and Assumptions 158
4.4.1.4 Separation of Flows and Definition of Submodules 159
4.4.1.5 Building Modeling Scenarios 159
4.4.1.6 Specification of the Modeling Context and the Genericity Degree 160
4.4.1.7 Identification of Decisions related to Modules 161
4.4.1.8 Definition of Module KPIs 161
4.4.1.9 Specification of the Scale Level 161
4.4.1.10 Definition of Departments linked to Modules 162
4.4.1.11 Definition and Association of Information of Modules with MS feature 162
4.4.1.12 Identification of the Raw Data and per-Department Organization 163
4.4.1.13 Identification of Elements Involved in Each Module 164
4.4.2 Design Phase: Instantiating Metamodels of Views 165
4.4.2.1 Decision of the Modeling Choices of Each Submodule . . . 165
4.4.2.2 Choice of the Modeling Approach of Each Submodule . . . 165
4.4.2.3 Specification of Function-Triggering Events 166
4.4.2.4 Decisions Impacted by Update Events 166
4.4.2.5 Specification of the Events Impacting on Each Information 167
4.4.2.6 Building the Structure and Format of KPIs 167
4.4.2.7 Data Structures 167
4.4.2.8 Drawing an Organogram 168
4.4.2.9 Instantiation of the Metamodels of Views 169
4.4.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module 169
4.4.3.1 Choice of the Technical Solution to Capture the Raw Data 170
4.4.3.2 Creation of the Database Tables Required by the Data Model 170
4.4.3.3 Creation of the Data Access Layer 171
4.4.3.4 Building the Business Model Layer 171
4.4.3.5 Creation of the Mapping Layer 171
4.4.3.6 Writing of the Pseudo-Code of Modules 172
4.4.3.7 Building an Optimization Model or a Decision Tree 172
4.4.3.8 Choosing the Solving Approach 173

4.4.3.9 Conducting Numerical Experiments . . . 174
4.4.3.10 Building a Simulation Model of Modules . . . 174
4.4.3.11 Mapping of the Decision Variables of the Optimization Model with the Simulation Model Attributes . . . 175
4.4.3.12 Creation of Traceability Files . . . 175
4.4.3.13 Launching the Experimentations . . . 176
4.4.3.14 Running the Corner Tests . . . 176
4.4.3.15 Displaying the KPIs . . . 176
4.4.3.16 Checking the Module Agility and Flexibility . . . 176
4.4.4 Maintenance Phase: Versioning of the Model of Each Module and Generation of the Corresponding Documentation . . . 177
4.4.4.1 Documentation and Versioning of Module Models . . . 177
4.4.4.2 Checking the Compatibility Among Views Models of Modules . . . 177
4.4.4.3 Organization of Review Meetings with the Project Team . 177
4.4.4.4 Updating Documentation and Versioning of Models of Modules . . . 178
4.5 Application of MEMO I4.0: Operational Stage . . . 178
4.5.1 Definition Phase . . . 178
4.5.2 Design phase . . . 178
4.5.3 Implementation phase . . . 179
4.5.4 Maintenance Phase . . . 180
4.6 Conclusion . . . 180

5 Application of MEMO I4.0 to Real Industrial Environment: a Case Study 183
5.1 Introduction . . . 184
5.2 Strategical/Tactical Stage: Development of Modules . . . 185
5.2.1 Definition Phase: Requirements and Specifications of Modeling . . . 187
5.2.1.1 Definition of Modeling Objectives and Scope . . . 187
5.2.1.2 Description of the Enterprise Vision, Guidelines & DNA . 187
5.2.1.3 Decomposition into Modules and Definition of their Functional Description and Assumptions . . . 189
5.2.1.4 Separation of Flows and Definition of Submodules . . . 191
5.2.1.5 Building Modeling Scenarios . . . 191
5.2.1.6 Specification of the Modeling Context and the Genericity Degree . . . 193
5.2.1.7 Identification of Decisions related to Modules . . . 193
5.2.1.8 Definition of Module KPIs . . . 194
5.2.1.9 Specification of the Scale Level . . . 194

5.2.1.10 Definition of Departments linked to Modules 195
5.2.1.11 Definition and Association of Information of Modules with MS feature 195
5.2.1.12 Identification of the Raw Data and per-Department Organization 196
5.2.1.13 Identification of Elements Involved in Each Module 196
5.2.2 Design Phase: Instantiating Metamodels of Views 197
5.2.2.1 Decision of the Modeling Choices of Submodules 197
5.2.2.2 Choice of the Modeling Approach of Each Submodule . . . 198
5.2.2.3 Specification of Function-Triggering Events 199
5.2.2.4 Decisions Impacted by Update Events 200
5.2.2.5 Specification of the Events Impacting on Each Information 200
5.2.2.6 Building the Structure and Format of KPIs 201
5.2.2.7 Data Structures 201
5.2.2.8 Drawing an Organogram 201
5.2.2.9 Instantiation of the Metamodels of Views 202
5.2.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module 205
5.2.3.1 Choice of the Technical Solution to Capture the Raw Data 206
5.2.3.2 Creation of the Database Tables Required by the Data Model 206
5.2.3.3 Creation of the Data Access Layer 206
5.2.3.4 Building the Business Model Layer 207
5.2.3.5 Creation of the Mapping Layer 208
5.2.3.6 Writing of the Pseudo-Code of Modules 208
5.2.3.7 Building and Solving an Optimization Model, and Conducting the Numerical Experiments 209
5.2.3.8 Building a Simulation Model of Modules 209
5.2.3.9 Creation of Traceability Files 212
5.2.3.10 Launching the Experimentations 212
5.2.3.11 Running the Corner Tests 213
5.2.3.12 Displaying the KPIs 214
5.2.3.13 Checking the Module Agility and Flexibility 214
5.2.4 Maintenance Phase: Versioning of the Model of Each Module and Generating the Corresponding Documentation 215
5.2.4.1 Documentation and Versioning of Module Models 215
5.2.4.2 Checking the Compatibility Among View Models of Modules 215
5.2.4.3 Organization of Review Meetings with the Project Team . 216

5.2.4.4 Updating Documentation and Versioning of Models of Modules . . . 216
5.3 Operational Stage: Investigating of the Internal Logistic by Simulation . . 216
5.3.1 Definition Phase . . . 217
5.3.2 Design Phase . . . 217
5.3.3 Implementation Phase . . . 218
5.3.4 Maintenance Phase . . . 219
5.4 Conclusion . . . 220

Conclusion & Perspectives (french version) 223

Conclusion & Perspectives 231

References 237

A Publications 275

B MS Features 277

C Conceptual Models 291
C.1 Conceptual Models of Views: Metamodels . . . 291
C.1.1 Conceptual Model of Partial Integration of Views of an MS . . . 291
C.1.2 Conceptual Model of Process View of an MS . . . 292
C.1.3 Conceptual Model of Data View of an MS . . . 293
C.1.4 Conceptual Model of Resource & Organization View of an MS . . . 295
C.1.5 Conceptual Model of Integration of All Views of an MS . . . 297
C.2 Conceptual Models of MEMO I4.0 Principles . . . 299

D Examples of Modeling Approaches 303
D.1 Independent Approach: Example of Modeling the Line Feeding Policy . . . 303
D.2 Intermediate Approach: Example of Modeling the Storage Strategy . . . 304
D.3 Integrated Approach: Example of Modeling the Production Policy . . . 305

E Application of MEMO I4.0: Didactic Example of Entreprise E1 307
E.1 Roadmap of Implemntation of MEMO I4.0 (detailed version) . . . 307
E.2 Definition Phase of MEMO I4.0: Didactic Example of E1 . . . 309
E.2.1 Assumptions related to the Modules of the Production Activity of E1 309
E.2.2 Choice of Scale level of Assembly Module of E1 . . . 309
E.2.3 Modeling Approach of Physical Submodule of Assembly Module of E1 . . . 311
E.3 Design Phase of MEMO I4.0: Didactic Example of E1 . . . 313
E.3.1 Modeling Choices of Each Submodule of E1 Modules . . . 313

E.3.2 Instantiation of the Function View Metamodel of E1 315
E.4 Implementation Phase of MEMO I4.0: Didactic Example of E1 316
E.4.1 Mapping of the Decision Variables of the Schedule Module of E1 with Attributes of the Simulation Model 316
E.4.2 List of Notations Used in the Pseudo-Codes 318
E.4.3 List of Algorithms (the Pseudo-Codes) 319

F Application of MEMO I4.0: elm.leblanc company 325
F.1 Definition Phase 325
F.1.1 List of the Modeling Scenarios of Modules of the Project MEMO I4.0 within elm.leblanc 325
F.1.2 List of Tables related to the Function Data of Modules 333
F.1.3 Identification of the Raw Data and its Organization per Department 338
F.2 Design Phase 338
F.3 Implementation Phase 345
F.3.1 Creation of the Data Access Layer 345
F.3.2 List of Algorithms (the Pseudo-Codes) 346
F.3.3 Building a Simulation Model of Modules: Results 350
F.3.4 Sensitivity Analysis for Variables: X1 and X2 352
F.4 Maintenance Phase 354
F.4.1 Documentation and Versioning of Module Models 354
F.5 Operational Stage: What-if Analysis for the Increase of Customer Demand 358
F.5.1 Definition Phase 358
F.5.2 Design Phase 358
F.5.3 Implementation Phase 359
F.5.4 Maintenance Phase 359
F.6 Operational Stage: Coupling the Simulation and Optimization for Addressing the ALBP with KPI 4.0 (Ergonomics) 360
F.6.1 Definition Phase 360
F.6.2 Back to the Development Cycle 360
F.6.2.1 Definition Phase of Strategical/Tactical Stage 361
F.6.2.2 Design Phase of Strategical/Tactical Stage 361
F.6.2.3 Implementation Phase of Strategical/Tactical Stage 362
F.6.2.4 Maintenance Phase of Strategical/Tactical Stage 367
F.6.3 Design phase 367
F.6.4 Implementation phase 367
F.6.5 Maintenance phase 369
F.7 Instantiation of Models 370

G List of utilities 377

Abstract **396**

Liste des Figures

1 Aperçu du manuscrit et des relations entre ses chapitres. 5
1.1 Exemple de structure de boîte SADT. 37
1.2 Exemple d'un diagramme de classe. 38
1.3 Exemples des types d'associations dans UML. 40
1.4 Synthèse des systèmes de production (Pour les dÃ©tails, voir Subsections 1.2.2 and 1.2.3). 53
1.5 Synthèse de modélisation d'entreprise. 54
1.6 Synthèse des concepts liés à la prise de décision dans les systèmes de production. 55
2.1 Changements de l'horizon temporel des problèmes de décision de l'ancien scénario au scénario Industrie 4.0. 91
3.1 Le périmètre d'étude couvert dans le manuscrit. 100
3.2 Le modèle conceptuel de la vue fonction d'un système de production. . . . 101
3.3 Éléments de structure de base d'une nouvelle procédure en trois étapes pour étudier le processus de décision dans les systèmes de production. . . . 102
3.3 Classification des décisions selon l'horizon temporel, la famille du problème, la branche du problème et l'activité. 104
3.4 Le modèle conceptuel de la vue décision d'un système de production. . . . 105
3.5 Le modèle conceptuel de la vue information d'un système de production. . 106
3.6 Le modèle conceptuel de la vue performance d'un système de production. . 106
3.7 Le modèle intégré des vues performance et information d'un système de production. 107
3.8 Interactions du problème d'affectation de zone. 108
3.9 Cartographie des décisions en fonction de l'horizon temporel et de l'activité. 120
3.10 Les impacts de la standardisation de l'échange des données sur l'horizon temporel des décisions. 123
3.11 Les impacts des nouvelles techniques et algorithmes sur l'horizon temporel des décisions. 125
3.12 Les impacts de la stratégie leagile sur l'horizon temporel des décisions. . . 126
3.13 Les impacts du sysème de production centré sur l'humain sur l'horizon temporel des décisions. 127

3.14 Les impacts du sysème de production adaptable sur l'horizon temporel des décisions. 129
3.15 Les impacts de la surveillance en temps réel sur l'horizon temporel des décisions. 130
3.16 Les impacts d'approche de contrôle décentralisé sur l'horizon temporel des décisions. 132
3.17 Les impacts de l'utilisation des nouveaux indicateurs clés de performances 4.0 sur l'horizon temporel des décisions. 133
3.18 Illustration de quelques évolutions depuis des interactions séquentielles à des interactions simultanées. 135
3.19 Illustration des nouvelles et potentielles intégrations de décisions. 136
4.1 Diagramme UML de l'usine intelligente. 141
4.2 Diagramme UML de la méthodologie de modélisation d'entreprise. 143
4.3 Les principes de modélisation d'entreprise dérivés depuis les principaux leviers de transformation. 144
4.4 L'approche structurée de MEMO I4.0. Le cycle de développement est l'élément clé dans la construction de la bibliothèque de modules. 146
4.5 Les objectifs de modélisation MEMO I4.0 dérivés depuis les principaux leviers de transformation. 148
4.6 Compatibilité entre les objectifs de modélisation et les dimensions du cadre de modélisation d'entreprise. 151
4.7 Modèle conceptuel illustrant les relations entre les défis et leurs principaux leviers transformations dans l'industrie 4.0, ainsi que les principes, les objectifs et les dimensions de MEMO I4.0. 152
4.8 Modèle conceptuel illustrant le concept de librairie de MEMO I4.0. 153
4.9 Modèle conceptuel illustrant les spécifications et objectifs de MEMO I4.0. . 154
4.10 Feuille de route pour l'implémentation de MEMO I4.0 (version compactée). 156
4.11 L'analyse SWOT pour guider la vision d'entreprise E1. 157
4.12 Exemple d'un diagramme SADT pour E1. 158
4.13 Illustration du principe de modularité par un exemple de découplage des flux (physique, information et contrôle) du module d'assemblage de E1. . . 159
4.14 Réutilisabilité et flexibilité des modules en fonction des dimensions: généricité et échelle. 166
4.15 Illustration du principe d'agilité à l'aide d'un exemple de configurations d'usine. 172
4.16 Approche pour construire le modèle de simulation. 174
4.17 L'architecture de développement du modèle de simulation intégré. 179
5.1 Les cas d'études au sein d'elm.leblanc traités par MEMO I4.0, ainsi que la chronologie du développement et de la capitalisation des modules. 186

5.2 L'analyse SWOT pour guider la vision de l'entreprise elm.leblanc. 188
5.3 SADT des processus du système de production. 190
5.4 Une capture d'écran de la base de données "*SmartFactoryDB*". 207
5.5 Le diagramme de probabilité des variables d'entrée ayant un impact sur le module Préparer. 211
5.6 Stock des composants dans les supermarchés. 220
C.1 Intégration partielle des vues. 291
C.2 Modèle conceptuel de la vue processus d'un système de production. 293
C.3 Modèle conceptuel de la vue données d'un système de production. 294
C.4 Modèle conceptuel de la vue resource & organisation d'un système de production. 296
C.5 Modèle conceptuel intégré d'un système de production. 298
C.6 Modèle conceptuel pour la dimension généricité. 299
C.7 Modèle conceptuel pour la dimension cycle de vie. 300
C.8 Modèle conceptuel pour la dimension vue. 301
C.9 Modèle conceptuel pour la dimension échelle. 302
D.1 Le niveau correspondant pour chaque politique d'alimentation de ligne. . . 303
D.2 Exemple d'approche intermédiaire pour modéliser des stratégies de stockage.304
D.3 Illustration des politiques de production flux poussé et flux tiré. Pour des raisons de simplification, le retour des postes de travail vers le MES est omis.305
D.4 Modélisation de la politique du flux poussé en utilisant la notion de signal d'un poste à l'autre. 306
E.1 Feuille de route pour l'implémentation de MEMO I4.0. 308
E.2 Exemple de niveau d'échelle utilisant le sous-module Ordre de fabrication. 310
E.3 Exemple de structure des DMS, FMS, et RMS. 312
E.4 Modèle de la vue fonction de tous les modules de E1. 315
F.1 Modèle de la vue fonction du module Préparer. 339
F.2 Modèle de la vue processus du module Préparer. 340
F.3 Modèle de la vue décision du module Préparer. 341
F.4 Modèle de la vue performance du module Préparer. 341
F.5 Modèle de la vue information du module Préparer. 342
F.6 Modèle de la vue données du module Préparer. 342
F.7 Modèle de la vue ressource & organisation du module Préparer. 343
F.8 Capture d'écran des procédures, des fonctions de table et des fonctions d'échelle définies dans la base de données. 345
F.12 Temps de blocage moyen par jour (gris) et utilisation de la capacité (bleu, rouge si critique) des postes de travail. 359
F.12 Composants critiques des postes de travail avec un temps de blocage moyen non nul. 359

F.13 Mise à jour du modèle de performance du module Produire. 362
F.14 Mise à jour du modèle de décision du module Produire. 363
F.15 % du temps de travail dans différents postes de travail pour SALBP et ILS solutions. 366
F.16 % de la charge de travail dans différents postes de travail pour SALBP et ILS solutions. 366
F.17 % d'utilisation de la capacité pour les solutions SALBP et ILS dans le scénario stochastique. 370
F.18 Instance du modèle fonctionnel du cas d'étude 1. 371
F.19 Instance du modèle processus du cas d'étude 1. 372
F.20 Instance du modèle décision du cas d'étude 1. 373
F.21 Instance du modèle performance du cas d'étude 1. 374
F.22 Instance du modèle information du cas d'étude 1. 374
F.23 Instance du modèle données du cas d'étude 1. 375
F.24 Instance du modèle ressource & organisation du cas d'étude 1. 376
G.1 Fonction utilisée par DAO pour obtenir les données à partir des procédures définies en base de données. 379
G.2 P-Valeurs du test Kolmogorov-Smirnov. 380
G.3 Classe Élément à créer dans la plateforme de simulation. 381
G.4 Créer un élément dans la plateforme de simulation (Partie 1). 382
G.5 Créer un élément dans la plateforme de simulation (Partie 2). 383
G.6 Créer un élément dans la plateforme de simulation (Partie 3). 384
G.7 Créer un élément dans la plateforme de simulation (Partie 4). 385
G.8 Créer un élément dans la plateforme de simulation (Partie 5). 386
G.9 Supprimer un élément dans la plateforme de simulation (Partie 1). 387
G.10 Supprimer un élément dans la plateforme de simulation (Partie 2). 388
G.11 Affectation des données selon le type d'élément. 389
G.12 Affectation des données à l'élément ensemble avancé. 390
G.13 Affectation des données à l'élément expression. 391
G.14 Affectation des données à l'élément file d'attente. 392
G.15 Affectation des données à l'élément séquence. 392
G.16 Affectation des données à l'élément ensemble. 393

List of Figures

1 Aperçu du manuscrit et des relations entre ses chapitres. 5
1 Overview of the manuscript and of the relations among its chapters. 10

1.1 Example of SADT box structure. 37
1.2 Example of class diagram. 38
1.3 Examples of the association types in UML. 40
1.4 Summary of manufacturing systems (for details see Subsections 1.2.2 and 1.2.3). 53
1.5 Summary of Enterprise Modeling. 54
1.6 Summary of concepts related to the decision making in Manufacturing Systems. 55

2.1 Changes of the time horizon of decision problems from the old scenario into Industry 4.0 scenario. 91

3.1 The scope of study covered in the manuscript. 100
3.2 The conceptual model of the function view of an MS. 101
3.3 Basic structure elements of a novel three-step procedure to study decision process in MSs. 102
3.4 The conceptual model of the decision view of an MS. 105
3.5 The conceptual model of the information view of an MS. 106
3.6 The conceptual model of the performance view of an MS. 106
3.7 The integrated model of the performance and information views of an MS. 107
3.8 Interactions of the zone assignment problem. 108
3.9 Cartography of decisions according to time horizon and activity. 120
3.10 The impacts of standardization of data exchange on time horizon of decisions. 123
3.11 The impacts of new techniques and algorithms on time horizon of decisions. 125
3.12 The impacts of leagile strategy on time horizon of decisions. 126
3.13 The impacts of human-centered MS on time horizon of decisions. 127
3.14 The impacts of adaptable MS on time horizon of decisions. 129
3.15 The impacts of real-time monitoring on time horizon of decisions. 130
3.16 The impacts of decentralization control approach on time horizon of decisions. 132

3.17 The impacts of using KPI 4.0 on time horizon of decisions. 133
3.18 Illustration of some evolution from sequential to simultaneous interactions. 135
3.19 Illustration of new and potential integration of decisions. 136

4.1 UML diagram of the smart factory 141
4.2 UML diagram of the EM methodology. 143
4.3 EM Principles derived from the main transformations drivers. 144
4.4 The structured approach of MEMO I4.0. The Development Cycle is the key element in building up the Library of Modules. 146
4.5 MEMO I4.0 modeling objectives, derived from the identified main transformation drivers. 148
4.6 Compatibility between modeling objectives and dimensions of EM framework. 151
4.7 Conceptual model illustrating relationships among challenges and their main transformation drivers in Industry 4.0, and the principles, objectives and dimensions of MEMO I4.0. 152
4.8 Conceptual model illustrating the concept of MEMO I4.0 library. 153
4.9 Conceptual model illustrating the MEMO I4.0 purpose and specifications. . 154
4.10 Roadmap for implementing MEMO I4.0 (compacted version). 156
4.11 The SWOT analysis to guide the enterprise vision E1. 157
4.12 Example of SADT diagram for E1. 158
4.13 Illustration of the modularity principle via an example of decoupling flows (physical, information, and control) of the assembly module of E1. 159
4.14 Reusability and flexibility of modules based on the genericity and the scale dimensions. 166
4.15 Illustration of agility principle using an example of plant configurations. . . 172
4.16 Approach to build the simulation model. 174
4.17 The Development architecture for the integrated simulation model. 179

5.1 The use cases within elm.leblanc handled by MEMO I4.0, and the chronology of modules development and capaitalization. 186
5.2 The SWOT analysis to guide the elm.leblanc vision. 188
5.3 SADT of MS process. 190
5.4 a screenshot of the "*SmartFactoryDB*" database. 207
5.5 The probability plot of the input variables impacting the Prepare module. 211
5.6 Component inventory in supermarkets. 220

C.1 Partial integration of views. 291
C.2 Conceptual model of the process view of an MS. 293
C.3 Conceptual model of the data view of an MS. 294
C.4 Conceptual model of resource & organization view of an MS. 296
C.5 Integrated conceptual model of an MS. 298

C.6 Conceptual model for the genericity dimension. 299
C.7 Conceptual model for the life cycle dimension. 300
C.8 Conceptual model for the view dimension. 301
C.9 Conceptual model for the scale dimension. 302

D.1 The corresponding level for each line feeding policy. 303
D.2 Example of intermediate approach to model storage strategies. 304
D.3 Illustration of push and pull production policies. For simplification reasons, feedback from workstations to MES is omitted. 305
D.4 Modeling the push policy using the concept of signal from a workstation to the next one. 306

E.1 Roadmap for implementing MEMO I4.0. 308
E.2 Example of scale level using the manufacturing order submodule. 310
E.3 Example of structure of DMS, FMS, and RMS. 312
E.4 Model of function view of the all modules of E1. 315

F.1 Model of the function view of the Prepare module. 339
F.2 Model of the process view of the Prepare module. 340
F.3 Model of the decision view of the Prepare module. 341
F.4 Model of the performance view of the Prepare module. 341
F.5 Model of the information view of the Prepare module. 342
F.6 Model of the data view of the Prepare module. 342
F.7 Model of the resource & organization view of the Prepare module. 343
F.8 Screenshot of procedures, table-valued functions, and scale-valued functions defined in database. 345
F.9 The box plot of the output variable Y1,Y2, and Y3 of the Prepare module. 351
F.10 Sensitivity analysis related to the X1 variable. 352
F.11 Sensitivity analysis related to the X2 variable. 353
F.12 Per-day mean blocking time (grey) and capacity utilization (blue, red if critical) of workstations. 359
F.13 Update of the performance model of the Produce module. 362
F.14 Update of the decision model of the Produce module. 363
F.15 % of working time in different workstations for SALBP and ILS solutions. . 366
F.16 % of workload in different workstations for SALBP and ILS solutions. . . . 366
F.17 % of capacity utilization for SALBP and ILS Solutions in the stochastic scenario. 370
F.18 Instance of the functional model of the use case 1. 371
F.19 Instance of the process model of the use case 1. 372
F.20 Instance of the decision model of the use case 1. 373
F.21 Instance of the performance model of the use case 1. 374

F.22 Instance of the information model of the use case 1. 374
F.23 Instance of the data model of the use case 1. 375
F.24 Instance of the resource & orgranization model of the use case 1. 376

G.1 Function used by DAO to obtain data from the procedures defined in database. 379
G.2 P-values of the Kolmogorov-Smirnov test. 380
G.3 Class of Element to create in simulation platform. 381
G.4 Create element in simulation plateform (Part 1). 382
G.5 Create element in simulation plateform (Part 2). 383
G.6 Create element in simulation plateform (Part 3). 384
G.7 Create element in simulation plateform (Part 4). 385
G.8 Create element in simulation plateform (Part 5). 386
G.9 Delete element in simulation plateform (Part 1). 387
G.10 Delete element in simulation plateform (Part 2). 388
G.11 Assignement of data according to element type. 389
G.12 Assignement of data to advanced set element. 390
G.13 Assignement of data to expression element. 391
G.14 Assignement of data to queue element. 392
G.15 Assignement of data to sequence element. 392
G.16 Assignement of data to set element. 393

Liste des Tables

1.1 Comparaison des types de systémes de production (DMS, FMS, RMS). . . 16
1.2 Caractéristiques les plus pertinentes des cadres de Modélisation d'Entreprise existants. 36
1.3 Stratégies possibles pour combiner les modèles de simulation et d'optimisation. 50
2.1 Les défis auxquels l'Industrie 4.0 doit faire face du point de vue de la prise de décision. 74
2.2 Évolution de l'horizon temporel de certaines décisions dans le système de production dans le contexte de l'Industrie 4.0. 90
3.2 Exemple de productivité d'un systémes de production. 107
3.3 Interactions du problème d'affectation de zonage illustrées par les propriétés CIAOC des problèmes de décision. 111
3.4 Interactions des problèmes de décision sur la base de la littérature. 115
4.1 Correspondance entre les principes de MEMO I4.0 et les objectifs de modélisation. 147
4.2 Correspondance entre les principes, les objectifs, et les dimensions de MEMO I4.0. 150
4.3 Séparation des flux des modules de l'enteprise E1. 159
4.4 Scénarios potentiels de transformation de type de système de production à considérer lors de la modélisation du module d'assemblage de E1. 160
4.5 Degré de généricité en fonction du contexte de modélisation. 160
4.6 Décision liée aux modules de l'activité de production de E1. 161
4.7 Indicateurs clés de performance de chaque fonction de l'activité de production de E1. 161
4.8 Départements impliqués dans chaque module de l'activité de production de E1. 162
4.9 Exemple d'informations liées au module ordonnancer de E1. 163
4.10 Éléments de E1 inclus dans chaque module. 164
4.11 Les événements qui déclenchent les modules de E1. 167
4.12 Exemple d'indicateurs clés de performance à évaluer pour l'activité de production de E1. 168

4.13 Structure de données du jeu de données des modules E1. 168
4.14 Les solutions techniques pour capturer les données brutes. 170
4.15 Attributs du fichier de traçabilité de chaque module de E1. 175
5.2 Flux de chaque fonction de l'activité du système de production. 192
5.3 Sous-modules correspondant aux modules. 192
5.4 Décisions liées à chaque module. 194
5.5 Indicateurs clés de performance de chaque fonction. 194
5.6 Départements impliqués dans chaque module en termes de gestion des données. 195
5.7 Éléments de l'usine inclus dans chaque module. 197
5.8 Les choix de modélisation des éléments impliqués dans les sous-modules du module Préparer. 199
5.9 Les principaux événements déclenchant chaque fonction. 200
5.10 Les principaux événements agissant sur les décisions envisagées dans chaque module. 200
5.11 Structure des indicateurs clés de performance du module Préparer. 201
5.12 Structure de données de l'ensemble de données du module de Préparer. . . 201
5.13 Solutions techniques pour capturer les données brutes. 206
5.14 Variables de plan d'expériences. 210
5.15 Les p-valeurs du test K-S pour les variables X1, X3, X5 du module Préparer. 210
5.16 Les p-valeurs de K-S pour les variables X3-1 et X3-2 indiquant le temps de préparation du support carton dans les zones de stockage 1 et 2, respectivement. 211
5.17 Attributs du fichier de traçabilité de chaque module. 212
5.18 Résultats des tests de simulation du module Préparer et comparaison avec les données historiques. 213
5.19 Versions de module et nombre des réunions de revue par module. 216
5.20 Toutes les versions du modéle de simulation du module Préparer. 217
5.21 Pourcentage d'utilisation par équipe de travail des préparateurs dans chaque zone de stockage. 219
5.22 Nombre de tournées par équipe de travail et leurs durées. 219
B.1 Abréviations et classes et catégories d'encodage. 278
B.2 Caractéristiques du processus. 279
B.3 Caractéristiques du produit. 280
B.4 Caractéristiques de la ressource. 281
B.5 Caractéristiques de l'effet externe. 282
B.6 Caractéristiques du coût. 283
B.7 Caractéristiques du temps. 284
B.8 Propiètés des problèmes de décision. 289

E.1 Hypothèses liées à l'activité de production de E1. 309
E.2 Les choix de modélisation pour les sous-modules des modules de E1. 314
E.3 Correspondance des variables de décision du module Ordonnoncer de E1 avec les attributs du modèle de simulation. 317
E.4 La signification des notations utilisées dans le pseudo-code (Partie 1). . . . 318
E.5 La signification des notations utilisées dans le pseudo-code (Partie 2). . . . 318
F.1 Scénarios à considérer pour modéliser le sous-module du processus de préparation. 325
F.2 Scénarios à considérer pour modéliser le sous-module ordre de préparation. 326
F.3 Scénarios à considérer pour modéliser le sous-module contrôle de préparation.326
F.4 Scénarios à considérer pour modéliser le sous-module de contrôle de d'équipe et ressource. 326
F.5 Scénarios à considérer pour modéliser le sous-module du processus de chargement. 326
F.6 Scénarios à considérer pour modéliser le sous-module du processus de transport. 327
F.7 Scénarios à considérer pour modéliser le sous-module ordre de ramassage des commandes. 328
F.8 Scénarios à considérer pour modéliser le sous-module contrôle de transport. 329
F.9 Scénarios à considérer pour modéliser le sous-module du processus de production. 330
F.10 Scénarios à considérer pour modéliser le sous-module ordre de fabrication. . 331
F.11 Scénarios à considérer pour modéliser le sous-module contrôle de production.332
F.12 Données de fonction du module Préparer et leurs caractéristiques associées du système de production. 333
F.13 Jeu de Données, et données de fonction du module Charger et leurs caractéristiques associées du système de production. 334
F.14 Jeu de Données, et données de fonction du module Transporter et leurs caractéristiques associées du système de production. 335
F.15 Jeu de Données, et données de fonction du module Produire et leurs caractéristiques associées du système de production. 336
F.16 Jeu de Données, et données de fonction du module Objet Commun et leurs caractéristiques associées du système de production. 337
F.17 Ensemble de données du module Préparer, leurs sources de données et les départements assurant leur gestion. 338
F.18 Résultats de la qualité de l'ajustement à l'aide du test de Kolmogorov-Smirnov pour les variables X1, X3 et X5. 350
F.18 Boîte à moustaches des variables de sortie Y1, Y2 et Y3 du module Préparer.351
F.18 Analyse de sensibilité relative à la variable X1. 352

F.18 Analyse de sensibilité relative à la variable X2. 353
F.19 Attributs de l'élément Atelier/Ligne. 354
F.20 Attributs de l'élément produit. 355
F.21 Attributs de l'élément composant. 355
F.22 Attributs de l'élément support de stockage. 356
F.23 Attributs de l'élément date. 357
F.24 Attributs de l'élément équipe. 357
F.25 Attributs de l'élément calendrier. 357
F.27 Correspondance des variables de décision du modèle d'optimisation avec les attributs du modèle de simulation du module Produire. 368
F.28 % de postes de travail à risque solutions SALBP et ILS. 369
F.29 Taux de satisfaction de la demande client pour les solutions SALBP et ILS. 369
G.1 Définition des choix de modélisation classiques. 378

List of Tables

1.1 Comparison of manufacturing systems types (DMS, FMS, RMS). 16
1.2 Most relevant features of existing EM frameworks. 36
1.3 Possible strategies for combining simulation and optimization models. . . . 50

2.1 The challenges that Industry 4.0 must face from a decision-making perspective. 74
2.2 Evolution of time horizon of some decisions in MS in the context of Industry 4.0. 90

3.1 Classification of decisions according to the time horizon, problem family, problem branch, and activity. 104
3.2 Example of MS productivity. 107
3.3 Interactions of zoning assignment problem illustrated by the CIAOC properties of decision problems. 111
3.4 Interactions of decision problems based on the literature. 115

4.1 Correspondance between MEMO I4.0 principles and modeling objectives. . 147
4.2 Correspondance between MEMO I4.0 principles, objectives and dimensions. 150
4.3 Flow separation for the enterprise E1 modules. 159
4.4 Potential scenarios of transformation of the MS type to consider when modeling the assembly module of E1. 160
4.5 Genericity degree in function of modeling context. 160
4.6 Decision linked to the modules of the production activity of E1. 161
4.7 KPI of each function of the production activity of E1. 161
4.8 Departments involved in each module of the production activity of E1. . . 162
4.9 Example of information related to the schedule module of E1. 163
4.10 Elements of E1 included into each module. 164
4.11 The events that trigger the modules of E1. 167
4.12 Example of KPIs to evaluate for the production activity of E1. 168
4.13 Data structure of data set of the E1 modules. 168
4.14 Technical solutions to capture raw data. 170
4.15 Attributes of traceability file of each module of E1. 175

5.1 Assumptions related to each module. . . . 191
5.2 Flows of each function of MS activity. . . . 192
5.3 Submodules corresponding to modules. . . . 192
5.4 Decisions linked to each module. . . . 194
5.5 KPI of each function. . . . 194
5.6 Departments involved in each module in terms of data management. . . . 195
5.7 Elements of plant included into each module. . . . 197
5.8 Modeling choices of the elements involved in the submodules of the prepare module. . . . 199
5.9 The main events triggering each function. . . . 200
5.10 The main events acting on decisions considered in each module. . . . 200
5.11 KPIs structure of the prepare module. . . . 201
5.12 Data structure of data set of the Prepare module. . . . 201
5.13 Technical solutions to capture data raws. . . . 206
5.14 Variables of design of experiments 210
5.15 P-values of K-S test for X1, X3, X5 variables of the Prepare module. . . . 210
5.16 P-Values of K-S for X3-1, and X3-2 variables indicating the preparation time of cardbbox support in storage area 1, and 2, respectively. . . . 211
5.17 Attributes of traceability file of each module. . . . 212
5.18 Result of the simulation tests of the Prepare module and comparison with historical data. . . . 213
5.19 Module versions and the number of meeting reviews per module. . . . 216
5.20 All versions of the simulation model of the Prepare module. . . . 217
5.21 Per-shift percentage usage of preparators in each storage area. . . . 219
5.22 Number of loops per shift and durations. . . . 219

B.1 Abbreviations and encoding classes and categories. . . . 278
B.2 Process features. . . . 279
B.3 Product features. . . . 280
B.4 Resource features. . . . 281
B.5 External effect features. . . . 282
B.6 Cost features. . . . 283
B.7 Time features 284
B.8 Properties of decision problems. . . . 286

E.1 Assumptions related to the production activity of E1. . . . 309
E.2 Modeling choices for submodules of E1 modules. . . . 314
E.3 Mapping decision variables of the Schedule module of E1 with attributes of the simulation model. . . . 317
E.4 The meaning of the notations used in the pseudo-code (Part 1). . . . 318

E.5 The meaning of the notations used in the pseudo-code (Part 2). 318

F.1 Scenarios to consider for modeling the preparation process submodule. . . 325
F.2 Scenarios to consider for modeling the preparation order submodule. 326
F.3 Scenarios to consider for modeling the preparation control submodule. . . . 326
F.4 Scenarios to consider for modeling the shift/resource control submodule. . 326
F.5 Scenarios to consider for modeling the loading process submodule. 326
F.6 Scenarios to consider for modeling the transport process submodule. 327
F.7 Scenarios to consider for modeling the picking order submodule. 328
F.8 Scenarios to consider for modeling the transport control submodule. 329
F.9 Scenarios to consider for modeling the production process submodule. . . . 330
F.10 Scenarios to consider for modeling the manufacturing order submodule. . . 331
F.11 Scenarios to consider for modeling the production control submodule. . . . 332
F.12 Function data of the Prepare module and their associated MS features. . . 333
F.13 Data set collection, and function data of the Load module and their associated MS features. 334
F.14 Data set collection, and function data of the Transport module and their associated MS features. 335
F.15 Data set collection, and function data of the Produce module and their associated MS features. 336
F.16 Data set collection, and function data of the Common Object module and their associated MS features. 337
F.17 Data set of the Prepare module, their data sources and departments ensuring their management. 338
F.18 Results of the goodness of fit using the Kolmogorov-Smirnov test for the variables X1,X3, and X5. 350
F.19 Attributes of the Workcenter/Line element. 354
F.20 Attributes of the product element. 355
F.21 Attributes of the component element. 355
F.22 Attributes of the storage support element. 356
F.23 Attributes of the date element. 357
F.24 Attributes of the shift element. 357
F.25 Attributes of the schedule element. 357
F.26 Critical components of workstations with non-zero mean blocking time. . 359
F.27 Mapping decision variables of the optimization model with attributes of the simulation model of the Produce module. 368
F.28 % of workstations in risk SALBP and ILS solutions. 369
F.29 Customer's demand filling rate for SALBP and ILS solutions. 369

G.1 Definition of classical modeling choices. 378

Liste des Algorithmes

1 Pseudo-Code de la Fonction Préparer. 208
2 Pseudo-Code de la Fonction Planifier. 320
3 Pseudo-Code de la Fonction Ordonnoncer. 321
4 Pseudo-Code de la Fonction Produire. 322
5 Alogorithme de Prédiction de la Demande. 323
6 Algorithme de Génération d'Ordre de Préparation. 346
7 Algorithme de Mise à Jour de la Liste de Collecte des Composants. 347
8 Algorithme de Collecte des Composants. 348
9 Algorithme de Gestion des Ordres de Collecte des Composants. 348
10 Algorithme d'Affectation des Transporteurs. 349

List of algorithms

1 Pseudo-Code of Prepare Function. 208
2 Pseudo-Code of Plan Function. 320
3 Pseudo-Code of Schedule Function. 321
4 Pseudo-Code of Produce Function. 322
5 Demand forecasting algorithm. 323
6 Algorithm of generation of Prepration Order. 346
7 Algorithm of updating the picking list 347
8 Algorithm of picking components 348
9 Algorithm of management of picking order 348
10 Algorithm of assignement of milkrun drivers 349

Introduction (french version)

Ce manuscrit décrit les travaux de recherche réalisés par l'auteur pendant sa thèse de doctorat, qui a eu lieu dans le cadre du programme CIFRE (Convention Industrielle de Formation par la REcherche) entre Mines Saint-Etienne et l'entreprise elm.leblanc, filiale du groupe Bosch. Cette entreprise est un fabricant français de systèmes d'eau chaude, en particulier les chaudières. elm.leblanc est en train de mener une transition numérique de ses usines. Le Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes (LIMOS) a été le laboratoire d'accueil.

Au fil du temps, l'industrie a connu trois révolutions majeures, chacune a induit des changements techniques et organisationnels profonds. La dernière révolution industrielle, appelée "Industrie 4.0", a commencé en 2011 et est toujours en cours, dans le but de fournir aux machines de nouvelles capacités de communication avec leur environnement, de gestion de leurs états et d'amélioration autonome de leur fonctionnement. Les travaux de recherche décrits ici se concentrent sur la Modélisation d'Entreprise pour aider à la conception et à la gestion des Systèmes de Production, dans le but de développer des Systèmes d'Aide à la Décision adaptés au contexte de l'Industrie 4.0.

Étant donné que seules quelques entreprises ont signalé leur succès, et que, à notre connaissance, aucune d'entre elles ne fournit un plan de transformation, des efforts sont encore déployés pour proposer une feuille de route réaliste aux entreprises industrielles pour réussir leur transition numérique. Au cours de la dernière décennie, ces préoccupations ont suscité l'attention croissante des entreprises industrielles et du milieu universitaire, ainsi que l'intérêt de plusieurs pays en raison de leurs impacts sur différents aspects. Pour n'en citer que quelques-uns, l'aspect éducatif (par exemple, la création de nouveaux programmes de formation ou de partenariats entre les laboratoires de recherche et les entreprises industrielles), l'aspect de durabilité illustré par les trois piliers fondamentaux : économique, environnemental et social. Par exemple, la relocalisation des usines localement engendre d'un point de vue économique une réduction des coûts globaux, tandis que sur le volet environnemental, elle intègre la prise en compte de l'empreinte carbone dès la phase de conception des systèmes de production. Enfin, sur le plan social, cette initiative contribue á la création de nouveaux postes en réponse á l'émergence de profils et de compétences spécifiques. De nouveaux paradigmes ont donc émergé, entraînant des changements d'état d'esprit dans la

conception des systèmes de production, en particulier dans le contexte de l'Industrie 4.0. Logiquement, la conception ou la réorganisation d'une usine peut être considérée comme le résultat d'une successions de décisions prises. D'autre part, la construction d'une usine intelligente implique également de multiples décisions présentant plusieurs défis qui doivent être abordés par de nouvelles générations de systèmes d'aide à la décision. Ces derniers sont indispensables pour prendre des décisions dans de tels environnements si complexes.

Pour toutes ces raisons, ce manuscrit étudie les problèmes de décision dans les systèmes de production et comment ils sont impactés par les changements induits par l'Industrie 4.0 dans la prise de décision. Pour être compatible avec ces changements, il faut identifier les caractéristiques nécessaires d'une méthodologie de modélisation d'entreprise. En résumé, les objectifs de ce travail sont :

- Fournir une taxonomie et une cartographie des problèmes de décision dans les systèmes de production,
- Identifier les changements dans la prise de décision au sein de l'Industrie 4.0,
- Proposer une nouvelle méthodologie de modélisation d'entreprise adaptée à l'Industrie 4.0,
- Appliquer la méthodologie de modélisation d'entreprise proposée pour fournir une nouvelle génération des systèmes d'aide à la décision adaptés à l'Industrie 4.0.

Ce manuscrit est organisé en cinq chapitres, qui sont décrits ci-dessous et illustrés dans la Figure 1.

Chapitre 1 présente une vue d'ensemble des systèmes de production au lecteur, ainsi qu'un aperçu de leur évolution depuis le XVIIIe siècle jusqu'à l'Industrie 4.0. Il aborde les paradigmes industriels des différentes époques, notamment la production artisanale, la production de masse, la production lean et la personnalisation de masse, et enfin, la production axés sur l'innovation. Il présente les transformations significatives induites par la transition vers l'Industrie 4.0, notamment la disponibilité des données et la communication permanente entre les composants du système. Il examine d'abord le rôle des cadres de modélisation d'entreprise existants dans la littérature pour dériver des systèmes d'aide à la décision, en mettant l'accent sur ceux spécifiquement conçus pour l'Industrie 4.0. Ensuite, il introduit les systèmes d'aide à la décision et montre leur importance dans les systèmes de production, ainsi que leurs types et leur évolution dans le contexte de l'Industrie 4.0. Enfin, le **chapitre 1** présente les défis imposés par le monde VUCA (Volatilité, Incertitude, Complexité, Ambiguïté) et la nécessité d'une prise de décision proactive dans les systèmes de production.

Chapitre 2 donne un aperçu des problèmes de décision dans les systèmes de production et de leur évolution dans l'environnement de l'Industrie 4.0. Il commence par identifier les *problèmes de décision* existants dans les systèmes de production, en les répartissant dans trois principales *activités d'usine* : Production, Approvisionnement et Stockage. Il discute ensuite les leviers de transformations significatifs depuis le scénario industriel avant l'avènement de l'Industrie 4.0 vers *l'usine intelligente*, en identifiant trois principales *évolutions* : techniques, organisationnelles et transversales. Enfin, le **chapitre 2** met en évidence l'évolution des problèmes de décision dans les systèmes de production dans le contexte de l'Industrie 4.0, y compris les changements dans l'horizon temporel de certaines décisions (par exemple, la prise de décisions en temps réel), les changements des types de systèmes de production.

Chapitre 3 propose une taxonomie des problèmes de décision dans les systèmes de production, dans le but de définir un outil de cartographie qui s'applique au contexte de l'Industrie 4.0, afin d'approfondir le processus de prise de décision dans les systèmes de production dans cet environnement complexe. Le chapitre propose d'abord une procédure en trois étapes pour étudier la prise de décision dans les systèmes de production, ainsi que les interactions entre les décisions dans les systèmes de production. Par la suite, une taxonomie des problèmes de décision identifiés dans les systèmes de production est proposée, basée sur quatre facteurs : l'horizon temporel, la famille du problème, la branche du problème et le type d'activité. Pour chaque problème de décision identifié, cinq propriétés sont détaillées, à savoir : les Contraintes, les Entrées, les Hypothèses, les Sorties et les Critères, sur la base desquels un formalisme est introduit pour modéliser les interactions entre les décisions, ainsi que pour illustrer les changements survenus dans l'Industrie 4.0. L'outil de cartographie proposé, illustrant les problèmes de décision et leurs interactions, peut être également utilisé pour étudier plus en profondeur l'évolution de la prise de décision dans les systèmes de production en Industrie 4.0 étudiée dans le **chapitre 2**.

Chapitre 4 commence par introduire les concepts de base de la modélisation des systèmes de production et l'évolution de la modélisation d'entreprise dans l'Industrie 4.0, puis définit MEMO I4.0, une nouvelle méthodologie de modélisation d'entreprise adaptée au contexte de l'Industrie 4.0, capable de guider le processus de modélisation d'une usine intelligente. La motivation de créer MEMO I4.0 est que, à notre connaissance, les méthodologies de modélisation d'entreprise existantes n'adressent pas pleinement les défis du contexte de l'Industrie 4.0. L'analyse et les résultats des chapitres précédents servent de base à la définition de la nouvelle méthodologie de modélisation d'entreprise proposée, en mettant l'accent sur ses principes clés et ses dimensions. Une entreprise fictive est utilisée pour expliquer en détail les concepts de MEMO I4.0. Le chapitre se termine en résumant les applications potentielles de MEMO I4.0, notamment dans le développement de systèmes d'aide à la décision adaptés aux besoins des usines intelligentes.

Chapitre 5 se concentre sur l'application de MEMO I4.0 dans un environnement industriel réel présenté par une usine d'elm.leblanc. Pour prouver le temps de développement réduit grâce au mécanisme de capitalisation sur lequel MEMO I4.0 est basé, on considére un cas d'utilisation complexe. Le chapitre détaille l'application de MEMO I4.0 dans un scénario de modernisation (brownfield), ce qui signifie que l'usine en question existe déjà. Le cas d'utilisation est choisi parmi une série de cas d'utilisation identifiés par les managers et la direction de l'entreprise et impliquant le développement de nouveaux modules pour les systèmes de production. Ces modules sont ensuite ajoutés à une bibliothèque de modules pour une utilisation future. L'application de la méthodologie MEMO I4.0 est détaillée de bout en bout, en commençant par l'étape opérationnelle avec l'analyse des besoins des managers, puis en passant à l'étape stratégique/tactique pour le développement de nouveaux modules, et enfin en revenant à l'étape opérationnelle pour l'instanciation des modules et l'évaluation des performances. Le cas d'utilisation inclut l'étude de la logistique interne et l'impact de certains paramètres sur l'efficacité de l'usine.

Dans la section **Conclusion**, les points clés discutés dans le manuscrit sont résumés, et les orientations futures et les applications potentielles de ce travail sont exposées, ainsi que certaines pistes de recherche futures ouvertes par ce travail.

Deux autres cas d'utilisation sont présentés dans la partie **Annexe**. D'abord, un cas d'utilisation concernant l'augmentation de la demande des clients et la recherche de la configuration appropriée de l'usine, puis, un autre cas d'utilisation qui considére la prise en compte de nouveaux indicateurs de performance clés dans la phase de conception d'une ligne de production.

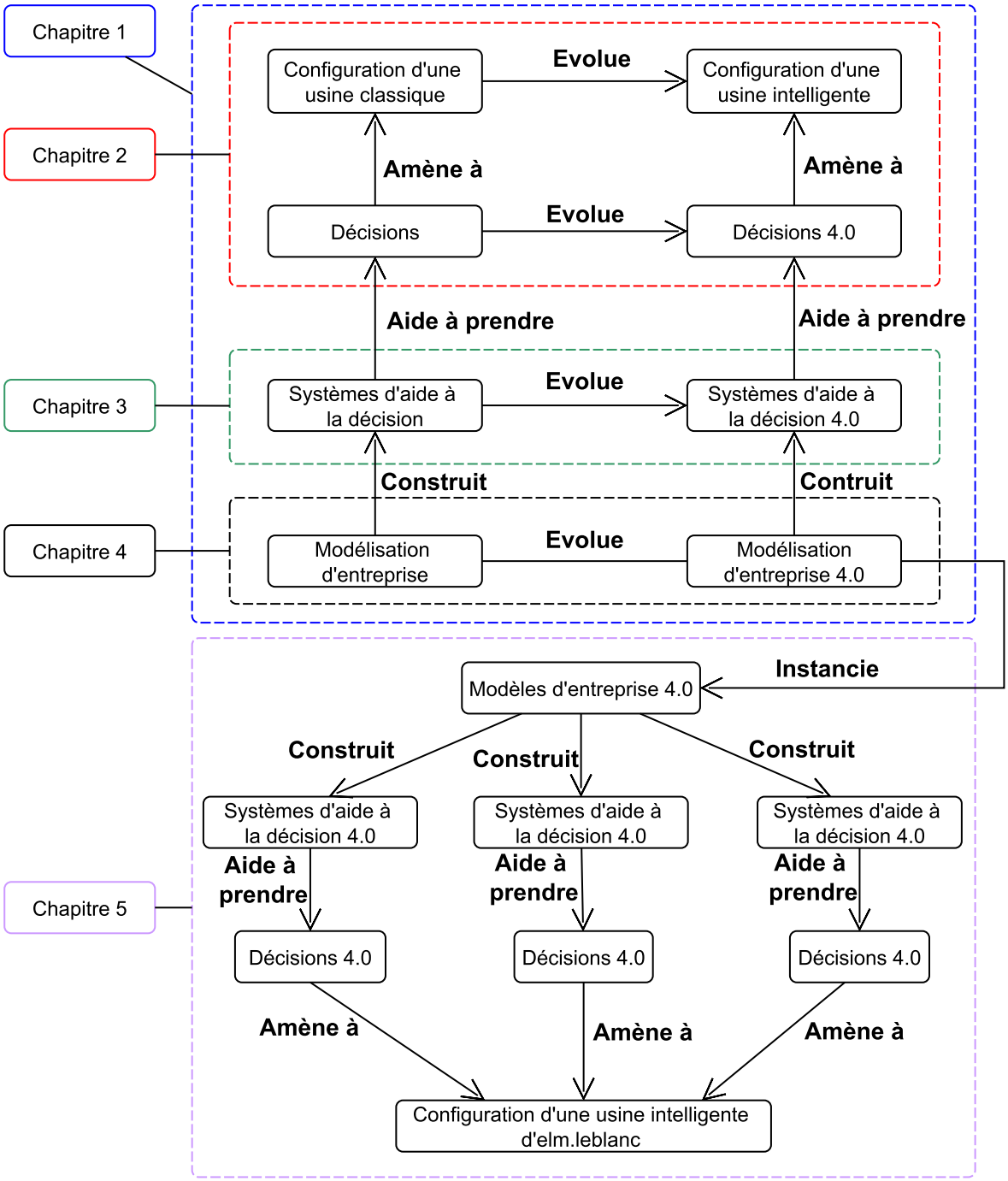

Figure 1: Aperçu du manuscrit et des relations entre ses chapitres.

Introduction

This manuscript describes the research work the author carried out during his doctoral thesis, which took place within the framework of the CIFRE (Convention Industrielle de Formation par la REcherche) program between Mines Saint-Etienne and the elm.leblanc company, a part of Bosch group. The company is a French manufacturer of hot water systems, particularly boilers, and is undergoing a numerical transition. The *Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes* (LIMOS) was the host laboratory.

Over the past, Industry has gone through three major revolutions, each one capable of inducing profound technical changes and organizational evolutions. The last industrial revolution, called "*Industry 4.0*", started in 2011 and is still ongoing, with the purpose of providing machines with new or augmented capabilities to communicate with their environment, manage their states, and autonomously improve their functioning. The research described here focuses on Enterprise Modeling (EM) to assist the design and management of Manufacturing Systems (MSs), with the aim to develop Decision Support Systems (DSSs) adapted to the Industry 4.0 context.

Since only a few companies reported their success, and to the best of our knowledge, none of them provides an exact footprint, efforts are still made to propose a realistic roadmap for industrial companies to succeed in their industrial transition. In the last decade, these concerns have gained the increasing attention of industrial companies and of academia, as well as the intereset of several countries due to their impacts on various aspects. To name a few of them, the educational aspect (e.g., the creation of new training programs or of partnerships between research laboratories and industrial companies), and the sustainability aspect represented by the Three Bottom Lines (TBL): economic, environmental and social. For instance, relocating factories locally not only reduces overall costs from an economic standpoint but also, from an environmental perspective, considers the carbon footprint since the design phase of MSs. Ultimately, on a social level, this initiative actively contributes to creating new job positions in response to the emergence of specific job profiles and competencies. New paradigms have consequently arisen, leading emerged mindset changes in the design of MSs, particularly in the Industry 4.0 context. Logically, the design or redesign of a plant can be seen as an outcome of multiple decisions that are organized based on their temporal sequence. On the other hand, constructing

a smart factory also involves multiple decisions presenting several challenges that must be addressed through new generation DSSs, which are much needed to make decisions in such complex environments.

For all of these reasons, this manuscript investigates the decision problems in MSs and how they are impacted by the changes in decision-making brought about by Industry 4.0. By doing so, it identifies the features an EM methodology requires to be compatible with these changes. To summarize, the objectives of the work are:

- Provide a taxonomy and cartography of decision problems in manufacturing systems,
- Identify the changes in decision-making within Industry 4.0,
- Propose a new enterprise modeling methodology adapted for Industry 4.0,
- Apply the proposed EM methodology capable to deliver new-generation DSSs adapted to Industry 4.0.

This manuscript is organized into five chapters, which are described in the following and depicted in Figure 1:

Chapter 1 presents a holistic view of MSs to the reader, as well as an overview of the evolution of MSs from the 18th century, to these days, i.e., to Industry 4.0. It discusses the industrial paradigms of the different epochs, including craft production, mass production, lean manufacturing, mass customization, and finally innovation-driven production. It discusses the significant transformations brought about by Industry 4.0 transition, including data availability and continuous communication among system components. It also investigates the role of existing EM frameworks in the literature for deriving DSSs, focusing on those specifically designed for Industry 4.0. Then, it introduces DSSs and their importance in MSs and gives their types and evolution within Industry 4.0. Finally, Chapter 1 presents the challenges imposed by the VUCA (Volatile, Uncertain, Complex, Ambiguous) world and the need for proactive decision-making in MSs.

Chapter 2 provides an overview of decision problems in MSs and how they are evolving in the Industry 4.0 environment. It begins by identifying existing *decision problems* in MSs in three main *plant activities*: production, supply, and storage. It then discusses the significant transformations from the industrial scenario before the advent of Industry 4.0 to the *smart factory*, identifying three main *evolutions*: technical, organizational, and transversal. Finally, Chapter 2 highlights the evolution of decision problems in MSs within Industry 4.0, including changes in the time horizon of some decisions (e.g., real-time decision-making), and changes in the types of MSs.

Chapter 3 proposes a taxonomy of decision problems in MSs, with the goal of defining a cratography tool which is applied to the Industry 4.0 context, so that decision-making

process in MSs in the Industry 4.0 context can be delved into a systematic way. First, it proposes a three-step procedure for studying decision-making in MSs, as well as the interactions between decisions in MS. A taxonomy of identified decision problems in MS is then proposed, based on four factors: time horizon, problem family, problem branch, and activity type. For each identified decision problem, five properties are detailed, namely: Constraints, Inputs, Assumptions, Outputs, and Criteria (CIAOC), based on which a formalism for modeling the interactions between decisions, as well as illustrating changes that occurred within Industry 4.0, is introduced. The mentioned cartography tool, illustrating decision problems and their interactions, can then be presented and used to illustrate and study more in depth decision-making evolution in MSs introduced in Chapter 2.

Chapter 4 starts by discussing the basic concepts of MS modeling and the evolution of EM in Industry 4.0, then defines MEMO I4.0, a new EM methodology adapted to the Industry 4.0 context and capable to guide the modeling process of a smart factory. The motivation to introduce MEMO I4.0 is that, in our opinion, existing EM methodologies do not fully address the challenges of Industry 4.0 context. The analysis and findings of the previous chapters serve as the foundation for the proposed new EM methodology, most importantly its key principles and dimensions. A fictitious enterprise is used to explain the concepts of MEMO I4.0 in detail. The chapter ends by summarizing the potential applications of MEMO I4.0, especially in the development of DSSs that fit smart factory needs.

Chapter 5 focuses on the application of MEMO I4.0 in the real industrial environment offered by an elm.leblanc plant by considering a complex use case and to prove the reduced development time thanks to the capitalization mechanism MEMO I4.0 is based on. The proposed EM methodology was applied in a brownfield scenario, meaning that an existing MSs was being updated. The use case is chosen among a set of use cases identified by the company's managers that involve the development of new modules for the MS. These modules are then added to a library of modules for future use. The application of the MEMO I4.0 methodology is detailed from strat to end, beginning at the operational stage with the analysis of the manager needs, then proceeding to the strategical/tactical stage for the development of new modules, and finally returning to the operational stage for the instantiation of the modules and the performance assessment. The use case includes investigating internal logistics and understanding the parameters that impact plant efficiency.

In the **Conclusion** section, the key points discussed in the manuscript are summarized, and future directions and potential applications of this work are highlighted, along with some potential avenues for future research opened up by this work.

Two other use cases are presented in the **Appendix**. A use case about increasing customer demand and finding the appropriate plant configuration, and another one

considering new Key Performance Indicators (KPIs) in the design phase of a production line.

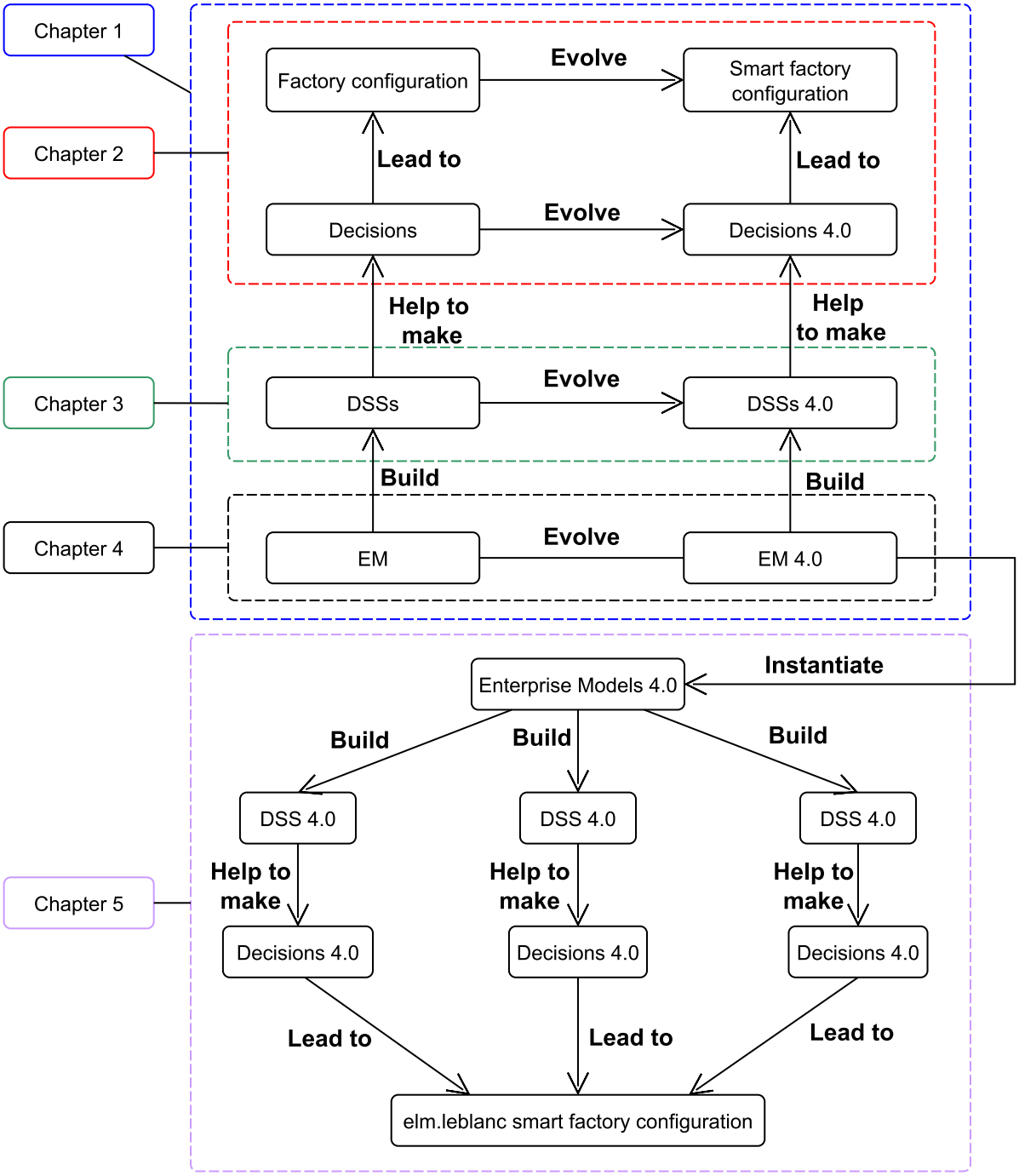

Figure 1: Overview of the manuscript and of the relations among its chapters.

Chapter 1

Decision Support Systems in Manufacturing Systems

Résumé

Ce chapitre présente une analyse de l'évolution des révolutions industrielles jusqu'à l'Industrie 4.0 actuelle. Chaque révolution a entraîné des changements dans les méthodes de production, passant de la production artisanale à la personnalisation de masse, et enfin à la production axée sur l'innovation en Industrie 4.0. L'Industrie 4.0 se caractérise par des systèmes de production interconnectés, basés sur les données, donnant naissance au concept d'usine intelligente. Ce chapitre donne lieu à trois questions de recherche pour explorer les problèmes de décision dans les systèmes de production à l'ère de l'Industrie 4.0. La première question vise à identifier les problèmes de décision existants et leur interactions pour améliorer la performance globale. La deuxième question porte sur l'impact de l'Industrie 4.0 sur les processus de prise de décision dans les systèmes de production. Enfin, la troisième question de recherche explore les fonctionnalités nécessaires d'un cadre de modélisation d'entreprise pour s'adapter aux changements de prise de décision induits par l'Industrie 4.0. Dans l'ensemble, ce chapitre établit les bases pour comprendre les défis et les opportunités de l'Industrie 4.0. Il souligne également l'importance des systèmes d'aide à la décision avancés pour réussir dans le monde complexe et volatile de l'Industrie 4.0.

Contents

1.1 Introduction 12
1.2 Manufacturing Systems: from Production to Information 14
1.2.1 Classification of Manufacturing Systems 15
1.2.2 Challenges of Manufacturing Systems in the Industry 4.0 17
1.2.3 Emergence of New Technologies in Manufacturing Systems 20
1.3 Enterprise Modeling Frameworks 28
1.3.1 Concepts and Terminology of Enterprise Modeling . . 28
1.3.2 The Role of Enterprise Modeling 32
1.3.3 Classical Frameworks of Enterprise Modeling 32
1.3.4 New Frameworks of Enterprise Modeling in Industry 4.0 34
1.3.5 Methods and Tools of Enterprise Modeling 36
1.4 Decision Making and Decision-Support in Manufacturing Systems 42
1.4.1 Importance of Decision-Support Systems in Manufacturing Systems 44
1.4.2 Types of Decision Support Systems 44
1.4.3 Quantitative Modeling 45
1.4.4 Decision Making and Decision Support Systems in the Industry 4.0 49
1.5 Open Research Questions 50

1.1 Introduction

In the 18th century, the First Industrial Revolution occurred, mechanical systems and steam power started to be standardized. The Second Revolution emerged at the beginning of the 20th century, thanks to the introduction of electricity. Around the 1970s, the Third Revolution started with computers and automation systems development (Hermann et al., 2016). In the last decade, the transition to the Fourth Revolution started and is still ongoing, which focuses on the connectivity of the global system of the enterprise using the internet technologies into industry (Liu and Xu, 2016).

The four industrial revolutions were initiated by market development, societal needs, and new technologies development and have been accompanied each time by different production organizational paradigms (Koren, 2010). The first paradigm was the craft production known to be traditional and manual, and its production process was specifically designed to create only on one product type (Herrmann et al., 2014). Mass production was the second paradigm that Ford Motor Company initiated to produce a large quantity of standardized products. Afterward, the lean manufacturing paradigm was developed by Taiichi Ohno and implemented by Toyota Motor Company after the

oil crisis of 1973. The objective was to eliminate all waste in the production process and increase efficiency without additional resources (Bhamu and Singh Sangwan, 2014).The guidelines proposed by the Toyota Production System were and still are the reference for many companies to succeed in the set-up of lean manufacturing (Shah and Ward, 2007). This paradigm coincided with the third revolution, which introduced automation in various industries, and enabled the development of automated systems and machines that could perform tasks previously done by humans. Automation also led to the creation of new jobs in fields such as computer programming, engineering, and information technology. A few years later, between the late 1980s and early 1990s, the need for advanced business systems like Enterprise Resource Planning (ERP) and Manufacturing Executive System (MES) emerged, marking a significant evolution of MSs during the third industrial revolution. These business systems played a crucial role in enhancing maturity within MSs and, subsequently, the apparition of a new paradigm called mass customization. This latter aims to provide a diversity of products tailored to the needs of customers, and is supported by the use of lean practices. These practices are seen as effective ways to manage the uncertainties that come with mass customization (Abdallah and Yoshiki, 2009). (Stanley, 1987) was among the first to popularize the mass customization and (Pine, 1993) further introduced the concept. This latter argues that to transition from mass production to mass customization, a company must shift its focus from operational efficiency to process efficiency and continuously improve its production systems to meet customer needs.

Finally, the fourth industrial revolution called *Industry 4.0* was introduced at the Hannover Messe trade fair (Drath and Horch, 2014), and was associated with the *factory of the future* or the *smart factory*. This latter emphasizes the connection of all production system components (e.g., products, machines, tools, operators, and the external environment) to achieve agility in quickly solving problems arising in a factory (Radziwon et al., 2014), and leads to the interconnected smart factory thanks to data being available anytime and continuous communication. The organizational paradigm associated with the fourth revolution can be characterized as an innovation-driven production.

Although, the use of automation in smart factories is a natural evolution of the third revolution, nowadays, companies need not only the integration of advanced technologies such as *Internet of Things* (IoT), and *Artificial Intelligence* (AI), but also the integration of data into their decision process and ensure consistency with the business systems of the enterprise, evaluate the performance of their systems in near real-time, and optimize their configuration. In this new context, decision makers ask for a proactive way to make their company profit.

Companies are trying to adapt to the new paradigm of smart factories, in order to be more agile and stay ahead of their competitors in a VUCA (Volatile, Uncertain, Complex,

Ambiguous) world. Minciu et al. (2020) describe the VUCA environment as having a short product life cycle, rapid changes in processes, and an unpredictable future due to uncertainty in customer requirements and the complexity of organizational modes. The uncertainty mentioned in the VUCA world is broader compared to the uncertainty associated specifically with mass customization. Mainly, mass customization uncertainty refers to the uncertainty arising from the choices of customers in terms of volume, variety, and options, while uncertainty in a VUCA world encompasses a wider range of factors, such as political changes, rapid technological advances, market disruptions, geopolitical risks, and customer requirements.

Furthermore, since the development of manufacturing enterprises and their evolution, companies face many problems involving making complex decisions, such as investing in new technology to improve production efficiency. Decision makers must evaluate the cost and benefits of the new technology, determine the impact on the workforce, and ensure that the technology is compatible with existing systems. Researchers always tried to propose efficient ways to tackle these problems by correctly describing them and using the appropriate methods and tools to solve them.

In this chapter, we first define in Section 1.2 manufacturing systems and their types. Then we focus on the significant transformations induced by Industry 4.0. In Section 1.3, we present the role of existing enterprise modeling frameworks in the context of manufacturing systems. Then, we review classical enterprise modeling frameworks, the new ones proposed within Industry 4.0, and describe methods and tools commonly used in these frameworks. In Section 1.4, we introduce decision support systems, show their importance in manufacturing systems, and discuss their types and their evolution within Industry 4.0. Finally, Section 1.5 provides open research questions that will be addressed in this manuscript.

1.2 Manufacturing Systems: from Production to Information

Manufacturing is the backbone of industrialized society, triggered by the first industrial revolution (Jovane et al., 2008). Wu (1992) defined it as an organized activity to transform raw materials into finished goods. On the other hand, Sillitto et al. (2019) define a *system* as "*an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not*", and Chryssolouris (2006), describes a *manufacturing system* (MS) as "*a combination of humans, machinery, and equipment that are bound by a common material and information flow*".

An MS is the complex system used by a manufacturing enterprise to store raw materials, transport them to the workshop, and transform them into a final product in order to create their economic wealth. Resources and the exchange of information play a significant role

in the organization of an MS.

Sperandio (2005) identifies three main decompositions of MS: functional, organizational, and hybrid. The functional decomposition describes the MS functionalities sees an MS as a set of functions, processes, or activities. The organizational decomposition describes the MS architecture. It involves the resource structure, and the dynamics flow among these resources. Lastly, the hybrid decomposition takes into account both the functional and the organizational decompositions.

In this manuscript, to cover the whole plant activities, we consider a functional decomposition of MS that includes three activities: production, supply, and storage. At the same time, we will make use of an the organizational decomposition at the plant, line, workstation, or product level depending on the situation.

1.2.1 Classification of Manufacturing Systems

The typology of MS depends on either the final product types or the system structure.

With respect to product type, when MS produces discrete parts, we talk about discrete production (Chryssolouris, 2006), while continuous production, often known as the process industry, involves final products such as liquids, gases, or powders (Fransoo and Rutten, 1994).

In this dissertation, however, we classify MS depending on the system structure, as presented below, and consider the case of discrete production.

1.2.1.1 Manufacturing Systems Types

MSs can be divided according to their structure into three types, namely *Dedicated Manufacturing Systems* (*DMSs*), *Flexible Manufacturing Systems* (*FMSs*), and *Reconfigurable Manufacturing Systems* (*RMSs*).

DMS is a specialized factory. Koren et al. (2018) define it as a collection of dedicated machines specialized in producing only one product with a sequential arrangement. The transfer of product into DMS between machines may be synchronous or asynchronous (Edmondson and Redford, 2002). In synchronous transfer, the machines have to start and stop operating simultaneously, whereas with asynchronous transfer, machines are free to start or stop independently. This type of MS is often tailored to achieve high productivity with cost-effectiveness. Therefore, it is well-suited for mass production.

MSs evolved to FMSs for the first time in the 1980s to cost-effectively respond to three critical needs: customization, medium volumes, and medium variety (Shivanand et al., 2006). ElMaraghy (2005) defines an *FMS* as a built-in system combining computer numerical control (CNC) machines and handling equipment that work together under computer control.

The expansion of mass customization due to market fluctuations in volume and product type showed the limitations of FMS and led to a new type of MS. To react to the sudden changes of the market, Koren et al. (1999) introduced RMS as an MS capable of rapidly modifying its structure to adjust production capacity and functionality, i.e., product variety. The flexibility of RMS is not limited to its software components but also concerns its physical structure. The six main features of RMS are convertibility, scalability, customization, modularity, integrability, and diagnosability (Koren, 2006).

1.2.1.2 Comparison of Manufacturing Systems Types

Table 1.1 presents a brief comparison of the three MS types. The works of Abdi and Labib (2004); Delorme et al. (2013); ElMaraghy (2005); Koren (2006, 2010); Peters and McGinnis (1999) inspired our comparison.

Table 1.1: Comparison of manufacturing systems types (DMS, FMS, RMS).

	DMS	**FMS**	**RMS**
Trigger (Production paradigm)	Mass production	Lean manufacturing	Mass customization
Software part	Fixed	Adjustable	Adjustable
System structure	Fixed	Fixed	Adjustable
Product variety	Single product	Preplanned multiple products	New multiple products
Productivity	High	Low	Medium
Complexity	Low	High	Medium
Scalability	No	Limited	Yes
Adapted for market	Stable	Predictable	Uncertain
Responsiveness to market change	Not adapted	Slow	Rapid
Objective	Economy of scale	Economy of scope	Combination of the economy of scale and scope
Lifetime Cost	Low	High (can be Reasonable*)	Medium

First of all, the comparison shows that DMSs are simple and suitable for an economy of scale, but, inefficient with regard to products with a short life cycle and high demand variations. Moreover, the comparison highlights that the advantage of FMSs is their economy of scope, and their main drawback is the high initial investment cost (*the lifetime cost can be reasonable if production of many parts simultaneously at low volume). Finally, RMSs are adapted to cope with significant product design changes, although, their centralized control logic can be a critical point, especially in the new industrial era.

1.2.2 Challenges of Manufacturing Systems in the Industry 4.0

Industry 4.0 is a recent trend of the manufacturing industry, due to the development of new software and hardware technologies. In the literature, the term *Industry 4.0* is short for the Fourth Industrial Revolution. The term is widely used and has been first used in Germany (*Industrie 4.0*). However, there are many equivalent terms depending on the context, for example, *Industrie du Futur* in France and *Industrial Internet* in the United States.

The future vision of MSs calls for implementing modular solutions where products control their processes (Lasi et al., 2014). Implementation of Industry 4.0 includes three types of integration: vertical, horizontal and end-to-end (Brettel et al., 2014; Stock and Seliger, 2016).

- Vertical integration affects various MS subsystems and concerns equipping a factory with new technologies. The aim is to connect the physical world to its virtual twin and transform an MS into a smart factory (Wang et al., 2016).
- Horizontal integration is about connecting different smart factories to create collaborative value networks that provide products and services (Foidl and Felderer, 2015).
- End-to-end integration is about integrating the entire value chain (customers, suppliers, and all stakeholders) to offer customized products and services (Stock and Seliger, 2016).

While technological developments have transformed MSs, numerous challenges have emerged. Which need to be correctly identified in order to succeed in the transition towards the smart factory.

The short *product life cycle* (PLC) is one of the main challenges. Competition urges a rapid development of new products to gain new market shares. Besides, customers continuously ask for product diversity, which requires investing more time and money on product innovation (Lasi et al., 2014). However, to guide the process of product innovation, the prediction of customer needs is necessary. VUCA world of Industry 4.0 has complicated this prediction, and therefore, decisions about product development are becoming ambiguous. The smart factory requires developing smart and connected products, impacting product design principles and production process requirements.

The complexity of the process is another challenge, directly implied by the shorter PLC. The increase of product diversity with short PLC implies reconfiguring the MS more frequently. MS reconfiguration includes convertibility (product change) and scalability (production capacity). Companies stay competitive when their process can reduce time to market through MS reconfigurability. RMSs are the most competitive among existing

systems that offer this flexibility due to their main characteristics and ability to ensure a short ramp-up. Nevertheless, the centralized control system of RMSs is not compatible with the Industry 4.0 environment: the networked environment considers the industrial wireless network as the primary component (Wang et al., 2016) and recommends using a decentralized control system to obtain MS connectivity. Thus, to comply with Industry 4.0, either RMSs evolve, or new types of MSs are required. Connectivity improves the effectiveness of decision-making by considering multiple factors, but it also makes the implementation of the process more complex.

The integration of data into the MS business processes (data collection, processing, analytics, management among others) presents a significant challenge. Large amounts of data are accessible and ubiquitous in the MS through ICTs and sensors (Zheng et al., 2018). Furthermore, big data technology is undergoing rapid development to assist decision-makers in exploiting data and making accurate and timely decision-making. Industry 4.0 provides new opportunities for predicting future outcomes of MS by integrating real-time communication and big data analytics. Also, machine learning techniques pervade the MSs to increase the accuracy of prediction models and lead to systems capable of self-optimization. However, companies continue to face the challenge of integrating data with IT systems, mainly due to interoperability issues (e.g., managing heterogeneous data) and security issues (e.g., using cloud-based solutions). Industry 4.0 involves the convergence of IT and Operations Technology (OT) systems in MSs to create a connected and optimized ecosystem. IT systems support business processes with data-driven decision making, while OT systems handle real-time control and monitoring of manufacturing processes. Integration of advanced technologies enables real-time communication and data exchange, potentially leading to increased efficiency, reduced costs, and improved performance. Cybersecurity is crucial in Industry 4.0 due to increased connectivity and automation. It is also essential for the convergence of IT and OT. For this reason, manufacturers must implement encryption, software updates, access control, and incident response plans to mitigate cyber threats. In addition, a comprehensive approach to cybersecurity is necessary to ensure the security and reliability of MSs and protect the reputation of the business.

The development of sustainable MS is a critical challenge since the integration of the sustainability dimensions throughout the product life cycle, including the selection of materials, manufacturing processes, product usage, and disposal, is increasingly impacting customer product choices. The data-driven approach of Industry 4.0 helps to capture and analyze data such as energy consumption, carbon emissions, and ergonomics assessments, thus, leading to support the eco-design methodologies. By incorporating eco-design principles and leveraging data-driven decisions, the transition to a circular economy becomes smoother, improving overall product life cycle performance. Moldavska and Welo (2017) state that the definition of sustainable manufacturing is inconsistent in the

literature. However, in our dissertation, the term sustainability will be used to refer to all the three main sustainability aspects of an MS, namely economic, environmental and social, known as the Triple Bottom Line (TBL) (Elkington, 1997). The systematic review of Eslami et al. (2018) examines the question of sustainable MS and attempts to go beyond the TBL sustainability. Companies are aware of the need for the TBL aspects in the design and management of an MS. Moreover, in the light of Industry 4.0, companies seek to produce more products ergonomically with fewer materials and less energy. Therefore, it becomes necessary for the smart factory to integrate the sustainability dimensions as criteria to converge on sustainable smart MS. The European Commission announced the Industry 5.0 concept to highlight a value-driven revolution that includes societal needs: sustainability, resilience, and human-centric. Furthermore, several initiatives consider sustainability as a challenge covered by Industry 5.0 (Demir et al., 2019; Huang et al., 2022; Xu et al., 2021), and reduce Industry 4.0 to a technology-driven revolution. In our work, sustainability is assumed to be encompassed in the Industry 4.0 revolution. From our point of view, industrial companies cannot achieve the transition to the smart factory (see Section 1.1) without considering this challenge since customers pay more attention to these societal needs and privilege collaboration with eco-responsible companies. Also, our works, among others, integrated this dimension as a prerequisite before the apparition of the concept of Industry 5.0. For instance, Gonçalves Machado et al. (2019) provide some insights into achieving sustainable manufacturing in the Industry 4.0 context.

Another challenge which results from the previous one is the integration and development of new skills within an MS. First, Industry 4.0 asks for designing innovative products and highly flexible and sustainable processes. Then, the use of available data to control MSs efficiently. Therefore, decision-making must deal with increased complexity and uncertainty. Optimization and simulation techniques are proactive approaches to support decision-making and provide insights on possible alternatives. However, in order to make use of these techniques and develop optimization and simulation models (c.f. Subsections 1.4.3.2 and 1.4.3.3), companies need to hire new profiles that deeply understand the behavior of MSs and support decision-makers whenever needed. Pinzone et al. (2017) investigate the evolutions of job skills within Industry 4.0 w.r.t. five organization areas: Operation Management (OM), supply chain management (SCM), Product-Service Innovation Management, Data Science Management, and IT-operations technologies integration management.

Even though the evolution of business models in MSs is an asset of company competitiveness, it remains a challenge and even more so in the Industry 4.0 context. A business model describes the business logic of how a company creates, captures, and delivers value to customers (Osterwalder and Pigneur, 2010; Shafer et al., 2005). In the Industry 4.0 context, the innovation-driven production paradigm imposes a dynamic and interconnected business environment that, on the one hand, accelerates the technology

push, i.e., creating new needs and inciting customers to consume, and on the other hand, leads companies to change their mindset from product to service, i.e., shift from selling products to being capable of selling products, services or both. Concepts such as "Manufacturing-as-a-Service (MaaS)", "Product-as-a-Service (PaaS)", and "Factory-as-a-Service (FaaS)" are becoming popular business models. *MaaS* is a service that uses a digital platform to connect both the network of suppliers and customers to deliver products (Chaudhuri et al., 2021). In the case of *PaaS*, also known as a product-service system (PSS), the value creation remains on the integrated offering of products and services such as remote installation of maintenance (Baines et al., 2007). *FaaS* is an open manufacturing service based on ICTs that integrates the software and hardware systems. Its objective is to promote personalized production and provide a distributed factory as a type of service (Kang et al., 2018; Park et al., 2019). Companies resort to cloud computing and digital twin technologies to implement these business models and optimize value creation. These technologies create new forms of customer interaction, ultimately leading to unparalleled customer experiences.

Finally, the last challenge is uncertainty. Market complexity is increasing more and more, and due to new customer requirements for product customization and the globalization concept, customer demand has become more difficult to predict. Industry 4.0 context highlights this increasingly unpredictable demand, asking MSs for a capacity of adaptation that accelerates the decision processes even under uncertainty.

New technologies are an effective response to most of the previous challenges (Lu, 2017), but the use of different technologies and the coexistence of old and new machines can give rise to interoperability issues. Greenfield and brownfield scenarios within Industry 4.0 are ongoing concepts that focus on these issues. The greenfield scenario reflects the design of an MS from scratch and requires a high investment cost, while the brownfield scenario deals with the choice of retrofitting old equipment or replacing it with modern equipment (Müller et al., 2018).

The main objective for addressing the above challenges is implementing a smart factory and achieving the required agility. After identifying the challenges of MSs in Industry 4.0, we now focus on emerging new technologies within MSs and their impacts.

1.2.3 Emergence of New Technologies in Manufacturing Systems

Another essential aspect of Industry 4.0 is the usage of smart software and hardware technologies. Tao et al. (2019) identify the four core elements of the ICTs evolution: the Internet of things, cloud computing, big data analytics, and artificial intelligence, which we review in Subsection 1.2.3.1.Then, we will take a short look at two smart hardware technologies, that pave the way to new MSs capabilities: the collaborative robots, and additive manufacturing.

1.2.3.1 Software Technologies

Internet of things (IoT) consists in connecting smart devices (things) using an Internet-based solution. Such devices feature electronics and software to enable sensing and store electronic information (Atzori et al., 2010). We use the term Industrial Internet of Things (IIoT) for IoT application in Industry (Liao et al., 2018). The availability and low cost of smart devices such as Radio Frequency Identification (RFID) tags, IoT gateway, smart sensors, and wearable devices give rise to the widespread application of IIoT and is essential to the transition to Industry 4.0. The objective of IIoT is data acquisition and supervisory control. Bidirectional communication among smart devices, such as machine-to-machine (M2M), is a significant change to provide valuable information for decision-making (Schiele and Jan Torn, 2020). Even though several IoT communication protocols exist, such as Message Queuing Telemetry Transport (MQTT), which is a lightweight protocol for IoT devices and low-bandwidth networks, Constrained Application Protocol (CoAP), which is similar but more suited for devices with limited processing power and memory, or Hypertext Transfer Protocol (HTTP) that is the foundation of the World Wide Web and is used for web requests and responses, M2M mostly uses an OPC-UA (Open Platform Communications Unified Architecture) as an industrial communication protocol, as it is independent of the platform and operating system (Drahos et al., 2018). In this bidirectional communication, we use several technologies to uniquely identify objects and allow information exchange, such as Radio Frequency Identification (RFID), Barcode, and wireless communication. RFID employs radio waves to identify and automatically track tags attached to devices (Zhou and Shi, 2008), while, barcode technology uses a machine-integrated optical system to read data presentation (Gao et al., 2007). The well-known wireless communication methods are Wifi, Bluetooth, Zigbee, LoRa, and SigFox. For instance, Wifi and Bluetooth have higher data transfer rates, making them suitable for applications that require real-time data transmission, while Zigbee and LoRa are more suited for low-power, long-range applications. Lastly, Sigfox has the advantage of low-power consumption, but it comes at the cost of lower data transfer rates. IoT devices provide real-time data about the state of manufacturing resources and processes. This data can be used to build more accurate models of MSs, enabling manufacturers to make informed decisions by identifying areas for improvement, optimizing their operations, and reducing downtime.

Cloud Computing (CC) is a relatively recent IT service model that allows remote computing using the Internet (Yang and Tate, 2012). CC uses resources such as servers and data centers to perform computing operations. These resources are either physical or virtual machines. Furthermore, for cloud infrastructure, we can use either a single server architecture, where a single physical or virtual server provides all the computing resources or a shared pool of servers architecture, where the computing resources are

spread across multiple servers, providing a more scalable and resilient solution. In a shared pool of servers architecture, we manage the cloud infrastructure by a software layer that dynamically allocates resources based on demand, ensuring that each user has access to the computing power they need when needed. Current cloud solutions propose an infrastructure as a service, i.e., customers have access to virtual machines and do not need to buy resources (Manvi and Shyam, 2014). Also, these solutions offer a platform as a service, i.e., customers can manage their computational capacity easily and adapt it when required. Hence, customers are offered a more flexible and pay-per-use service. The well-known examples are the Azure platform of Microsoft, Amazon Web Services platform of Amazon, and App Engine platform of Google (Ferrer et al., 2016). Several types of cloud solutions exist depending on accessibility. For instance, public clouds are accessible using the Internet, while private clouds are only accessible via a private network (Li et al., 2017). IoT and CC technologies promote real-time data analysis and decision-making and ultimately improve productivity and quality since they deliver helpful information to smart devices and decision-makers (Yang et al., 2019). Cloud manufacturing is an industrial version of CC, transforming manufacturing resources into manufacturing services (Xu, 2012). However, according to Baily et al. (2013), MSs generate and store significant amounts of data, and do so in the industrial sector, more than any other sector, and this is even more true in Industry 4.0. Data stored in the cloud can be used to build more complex models of MSs and allow manufacturers to simulate various scenarios. Therefore, companies need techniques to handle the enormous data of cloud-based applications.

Big data is one of such techniques, allowing to manage high volumes of data (Xu and Duan, 2018). *Big data analytics (BDA)* collects, analyzes, manages, and exploits massive data from heterogeneous sources in various formats (structured, semi-structured, unstructured). Five dimensions, called 5V (volume, velocity, variety, veracity, and value), specify the main characteristics of big data (Yin and Kaynak, 2015). Volume depends on the amount of data generated, and velocity is related to the speed of producing data; variety refers to different data types, while veracity indicates data accuracy; value reflects the gain and impact from data. Classical database management tools are not appropriate anymore because they cannot process data with all the above-mentioned characteristics. In Industry 4.0, data gathered from an MS exhibit these characteristics since IoT generates large volumes of data from various smart devices and requires quick analyses in the cloud. Besides, different data sources in different formats need to be integrated to gain competitive assets. Collecting accurate and well-structured data is essential but not enough. Transforming raw data into information and then extracting valuable insights are essential tasks for decision-makers to improve MS. Patterns and insights identified by BDA can be used to improve the accuracy of models of MSs, which helps manufacturers to make more proactive decisions, such as predicting maintenance

needs or identifying opportunities for process optimization.

Artificial intelligence (AI) is the most important multi-faceted technology in the digital age (Borges et al., 2021). Usually, AI relies on big data as its source of information and the cloud as its operating platform. AI covers several areas such as reasoning, programming, expert systems, and machine learning (ML). To make decisions using classical AI algorithms, we define rules to simulate human behavior, whereas with ML, the algorithms aim to mimic human learning. The ML technique automatically extracts data patterns and acts without explicitly needing programming (Murphy, 2012). It is common to classify the learning process of ML into three classes: supervised, unsupervised, and reinforcement learning. Supervised learning learns from labeled data, which represents the training sets; unsupervised learning discovers patterns among the data and does not require labeled data; reinforcement learning focuses on learning from experience (Wuest et al., 2016). As reported by Borges et al. (2021), from the industry's perspective, the objective of AI is to assist the decision-making process. Using AI to support decision-makers becomes a necessity to meet the challenges of Industry 4.0, and optimize various decisions of MSs (e.g., production scheduling, inventory management, maintnenace control). Lee et al. (2018) use the term industrial AI to refer to AI in the industrial systems.

1.2.3.2 Hardware Technologies

A *collaborative robot*, or *cobot*, is a robot with the ability to work together with human operators. It can physically interact and share the same workspace. In this collaboration, robots are assigned repetitive tasks, so that human workers can focus on tasks with high add-value (e.g., solving complex decision-making) (Djuric et al., 2016). El Zaatari et al. (2019) define four different collaboration scenarios: independent, simultaneous, sequential, and supportive. In an independent scenario, the cobot and operator share the same workspace but work on separate workpieces. In a sequential scenario, they work together on the same workpiece with time dependencies between the process of the cobot and that of the human worker. In a simultaneous scenario, both work on the same workpiece and do so concurrently (at the same time) without the need for coordination or synchronization because they work on separate processes (two different tasks). Lastly, in the supportive scenario, they interact and work on the same process, on the same workpiece at the same time. In each of these scenarios, the deployment of cobots in the real world remains a complex task, as it involves not only technological difficulties but also social implications (Fast-Berglund et al., 2016). The collaborative robots necessitate considering when modeling MSs a new relationship among production resources to illustrate the interactions between a human operator and a robot.

In contrast to subtractive manufacturing, *Additive Manufacturing (AM)* integrates a set of technologies capable of joining materials to render objects from 3D model data,

typically layer upon layer (Gibson et al., 2010). The material could be either plastic or metal. Costabile et al. (2017) summarize the workflow of AM process. First, the creation of a Computer-Aided Design (CAD) model in STL format, then, the STL model is physically renderd layer by layer, and finally the piece is smoothned by treatment of surfaces. As identified by Lindemann et al. (2012), among AM advantages, there are flexible development, reduced time-to-market, and few spare parts in stock to manage. Its main disadvantages are the necessity of rework and the high cost of materials and machines. 3D printing is one of the AM technologies: it uses the inkjet-type printer heads to transform powder into the successive layers of plastic or metal, then uses adhesives to glue them together (Pragana et al., 2021). By creating complex parts and prototypes quickly and easily, AM helps accelerate the decision-making process, such as creating new products or modifying existing ones.

The development and implementation of these technologies allow for a stronger integration of the virtual and real world: the ultimate goal is to connect physical systems (e.g., MSs) with virtual models in computers, therefore achieving the connectivity asked by Industry 4.0.

In what follows, we will use the terms *real* and *physical world* interchangeably, and the same we will do with the terms *virtual*, *cyber*, and *digital world*.

1.2.3.3 New Technologies for Computerization of Manufacturing

Tao et al. (2019) state that the cyber-physical system and digital twin are promising technologies to enable cyber-physical integration. Likewise, augmented reality contributes in the computerization of manufacturing. We give a brief review of these technologies to understand better their role in Industry 4.0.

Augmented reality (AR) technology consists of adding virtual content to augment the real world. AR systems add a layer of visual information on the top of the physical world to facilitate collaboration with humans and access digital information (Kong et al., 2018). According to Wang et al. (2016), five components are essentials for AR system: sensor system, visualization technology, tracking system, user interface, and processing unit. On the one hand, by using cameras, the sensor system collects data from the environment, and visualization technologies enable the observation of digital information layered on top of the image of the real world. On the other hand, the positioning of digital objects precisely in the real world is possible by the tracking system. The user interface allows communication between the system and the user. The processing unit has a central role in executing the program to run the AR system (Egger and Masood, 2020). It analyzes the real-world environment and processes camera images to detect and track objects, surfaces, and gestures with the ultimate goal to create an immersive and interactive experience for the user. AR can be used to create digital twins of MSs, enabling manufacturers to simulate and test different scenarios in a virtual environment,

as well as providing real-time insights and data visualization.

Cyber-physical systems (CPSs) are made up of two parts, the physical system incorporating devices (e.g., sensors, actuators) and the cyber system including embedded computers and networks. The first part interacts with the physical world, while the second part allows to control it. A feedback loop between these two parts is required to ensure the integration of virtual and physical systems. This integration corresponds to the vertical integration defined in Subsection 1.2.2. Moreover, with the increasing complexity of processes, CPS has become considered the core of Industry 4.0 towards the smart factory (Wang et al., 2016). Sometimes, we use the term Cyber-physical Production System (CPPS) to refer to the interaction between physical and cyber worlds in production (Rodič, 2017).

Jiang (2018) proposes to rely either on ISA 95 or 5C architecture to build CPS for smart factories. ISA 95 is a standard developed by the International Society of Automation (ISA) that provides a framework for integrating enterprise and control systems in manufacturing environments, while 5C architecture is a framework for implementing Industry 4.0 solutions in manufacturing environments. We choose to go in depth into 5C architecture because it was proposed recently by Lee et al. (2015) specifically to develop and deploy CPS in Industry 4.0. The reader interested to investigate ISA 95 can refer to Unver (2012).

5C defines five levels, namely Connection, Conversion, Cyber, Cognition, and Configuration. Below, we enumerate them from the lower to the upper level.

1. The connection level is responsible for data acquisition, using smart devices (e.g., sensors and programmable logic controller), and data transfer to the IIoT gateway, using protocols (e.g., RFID and Bluetooth).

2. The conversion level performs the transformation of collected data into information by using algorithms. Depending on the computing resources required by the algorithms, the transformation can be carried out at the smart devices, at the IIoT gateway, or even at the cloud level.

3. The cyber level consists of gathering massive information from machine networks by means of an IIoT middleware that enables connectivity via standard IoT protocols. Then, information is forwarded to a data center on the cloud side, so that all collected information is transformed into knowledge using CC and efficient analytics methods (e.g., BDA and AI).

4. The cognition level displays knowledge extracted from the cyber level to the user for making decisions. We use different user interfaces to visualize information (e.g., Web user interface or AR user interface).

5. The configuration level is responsible for the feedback from the cyber system to the physical system. It carries out corrections to the physical system by adjusting the actuators (e.g., machine and cobot) at the connection level according to decisions made at the cognition level.

Each level plays a significant role in increasing the integration degree of the CPSs. The connection level leads to self-monitoring using IoT technology, while the conversion level brings self-awareness to machines. Concerning the cyber level, analytic methods provide machines with self-comparison and self-prediction. The cognition level seeks to outfit the MS with a self-diagnosis feature by aggregating knowledge from components and machines. Finally, the configuration level attempts to achieve a self-configuration and a self-adaptive MS based on feedback and coordination mechanism.

We focus on the 5C architecture to emphasize the distinction between data and information, illustrated by the connection and conversion levels. In terms of modeling, these levels represent two distinct yet related aspects. By describing the 5C architecture, we illustrate that resources become a data source communicated through messages using specific protocols. These data can be transformed into information and provide insights into the business since they correspond to functions that describe the MS behavior. The cognition level highlights the strong link between the information extracted and the performance assessment of a process. The cognition level also shows how the information extracted supports the decisions of users. Overall, the 5C architecture highlights the importance of leveraging data and information to optimize manufacturing processes, improve performance, and make informed decisions. This architecture will help us to specify the aspects to integrate into our modeling of MSs.

Digital twin (DT) is a technology that allows a virtual high-fidelity representation of the physical aspects of the real world such as processes, products, and devices (Zhuang et al., 2018). Such representation takes place abd ecolves at real-time, i.e., the digital copy changes and updates synchronously with its real counterpart (Lu et al., 2020). As it is the case for CPS, DT aims for the integration between the virtual and physical parts. It uses a simulation paradigm to predict the behavior of an MS in a virtual space and provides feedback to optimize it.

Schleich et al. (2017); Söderberg et al. (2017) consider a DT as a new paradigm of simulation and modeling (see Subsection 1.4.2 for modeling definition) because it includes the simulation of all phases of the product life cycle. In this regard, the company can leverage DT to ensure the end-to-end integration in Industry 4.0, leading to customize products throughout the value chain. Though the widespread implementation of IoT in smart factories enables data acquisition for developing a DT, the virtualization and synchronization with the real world require a cloud-based solution. In addition, BDA and AI applications may contribute to making a DT an intelligent and standalone entity (Malakuti et al., 2020). Recently, we observe DT applications in several large companies

such as Siemens, Bosch, General Electric, Dassault systems (Schleich et al., 2017). Using DT in MSs can help decision-makers predicting equipment (e.g., machine, smart devices) failures. In addition, it guarantees that at any time the equipment is used in such a way to get optimal performance by analyzing the continuous transfer of data (Tao et al., 2017).

1.2.3.4 Comparison of Cyber-Physical System and Digital Twin

According to Uhlemann et al. (2017), DT is a prerequisite for the development of CPS. Tao et al. (2019) analyze CPS and DT from multiple perspectives and discuss their similarities and differences, stating that both technologies will fundamentally change MSs and business models since they attempt to achieve the computerization of manufacturing. Both CPS and DTs have a physical and cyber part. However, for CPS, the physical part executes manufacturing functions, while the cyber part offers smart data management, analytics, and computing capabilities. In contrast, for DTs, virtual models are created to reproduce the geometries, physical properties, behaviors, and rules of the physical world. Consequently, the mapping relationship between the cyber and physical worlds is not the same for CPS and DT. To summarize, the DT can be used to simulate the behavior of physical assets and provide insights to support decision-making within a CPS, but on the other hand, the real-time data collected from CPS can be used to refine the virtual models of DT.

1.2.3.5 Conclusion on the Impact of New Technologies

New enabling technologies improve remarkably the potential of decision-making (Lu, 2017). They have changed the priority and rationality of decisions. For instance, in the past, the design of conveyor systems of an MS required heavy investments with a very long return on investment (ROI). Nowadays, companies can resort to modular solutions that can adapt to new material flow configurations: this provides material flow agility and allows much lower investments, especially when proposed as a service. Furthermore, workstations are more flexible in CPS with the help of smart devices and plug-and-play mechanisms, and can quickly adapt their operation processing and product sequences. Information is easily accessible and exploitable in near-real-time (Theorin et al., 2017). MS objects, mainly smart devices, are interconnected thanks to the IoT (Georgakopoulos et al., 2016). They could transmit a large quantity of data (e.g., historical data, current status, quality errors) to the cloud. Techniques such as BDA and AI can treat huge information and predict the behavior of an MS. Thereby, decision-makers can leverage technologies to make efficient decisions.

1.3 Enterprise Modeling Frameworks

As we have seen, MSs are typically complex systems and involve multiple processes and activities, including production, supply, and storage. Moreover, these activities are typically coordinated through a management structure, a defining feature of an *enterprise*, an organization engaged in commercial or industrial activity to make a profit.

In addition, an MS is subject to the same business risks and challenges as any enterprise, such as competition, market fluctuations, and supply chain disruptions. Therefore, an MS must be managed like an enterprise, carefully focusing on financial performance, risk management, and strategic planning.

For all these reasons, we propose to use the same modeling techniques for designing an enterprise to study MSs.

This section aims to present the most known frameworks in the literature that can be used to build enterprise models that are compatible with the enterprise business. Before describing such frameworks, we first define some basic concepts and terminology. Next, we emphasize the significant role of enterprise modeling in the context of MSs. We then proceed to review both the classical and modern frameworks of enterprise modeling and draw a comparison between them. Finally, we present a list of widely-used methods and tools in enterprise modeling.

1.3.1 Concepts and Terminology of Enterprise Modeling

A *Model* is a simplified representation or abstraction of reality (Turban et al., 2011). It reduces the complexity of reality and represents the system with the degree of abstraction required to solve problems by focusing only on relevant parts. *Modeling* involves transforming the real-world problem into a *model*. Kochikar and Narendran (1994) define it as "*the process of creating an abstraction of a real-world system which reflects the system properties to the desired degree of detail*".

The ISO 15704 norm ISO (2019) defines an *enterprise model* as "*a representation of an enterprise as well as entities within an enterprise, their interrelationships, their decomposition, and detailing to the extent necessary to convey what the enterprise intends to accomplish and how it operates*". As stated by Ostic and Cannon (1996), an example of a logical enterprise model could be the flowchart of the production process of a product. A dynamic representation of an enterprise may use a simulation model instead.

In ISO 19440 norm, an enterprise domain is defined as a group of activities managed by a single organization or a set of organizations working together. This manuscript considers a plant covering MS activities as an enterprise domain. Therefore, we model an enterprise as an entity composed of several plants and a plant formed by several MSs.

A structured approach is a set of stages required to design an enterprise model, or a part of it.

An enterprise architecture conceptualizes the company context through a framework of models reflecting different aspects of its reality. These models allow studying the enterprise and its environment and the principles guiding its design and evolution.

In enterprise modeling, a *building block* is a modular unit that represents a specific aspect or component of an enterprise. It helps break down the complex structure of an enterprise into smaller, more manageable parts for analysis and improvement. Building blocks are designed to be reusable and interchangeable, allowing them to be combined in different ways to create different enterprise architectures.

According to the ISO 15704 norm, *a framework* is "*a structure expressed in diagrams, text, and formal rules which relates the elements of an enterprise architecture to each other*".

(Kateel et al., 1996) differentiate between *modeling tools* and *modeling methods*. The former are techniques used to represent functions or activities of systems by using diagrams. In contrast, the latter are the "*guidelines to combine the modeling tools to model a particular system*".

Berio and Vernadat (2001) define *the enterprise modeling* as "*the art of developing models which accurately represent the structure and behavior of a business entity, i.e., a definite part of an enterprise*". A *methodology of enterprise modeling (EM)* defines a framework and a structured approach as well as possible modeling tools and methods to guide the building of a set of models, i.e., an enterprise architecture. The integration of these models gives rise to the comprehensive enterprise model.

It is common to define the structure of an EM framework using axes called *dimensions*. Typical examples of dimensions used in EM frameworks are life cycle, genericity, and model views (often shortened to *views*). Readers could refer to the ISO 15704 norm for the definition of the *life cycle* and the *genericity* dimensions, and to the ISO 19440 norm ISO (2020) for the definition of the view dimension.

The *life cycle* dimension defines a set of phases that allows structuring the modeling process, from the definition of model objectives and requirements to its implementation and maintenance. Each life cycle phase has its own objective and performs several steps to achieve it. Furthermore, the life cycle does not represent a time sequence but rather a logical sequence of steps.

The *genericity* concept represents the degree of reusability of the developed models w.r.t a change in the context. A *generic* model is usable in any context; a model valid only in a specific context is a *specific* model; lastly, a model is said to have a partial genericity degree, and therefore, is called a *partial* model, if it is in an intermediate case of reusability.

In order to better understand the genericity concept, let us consider the example of modeling a production policy (also called production control strategy). Three possible production policies exist, namely push, pull, and hybrid (Wang and Xu, 1997). A push policy consists of producing and pushing products to the market, while a pull policy produces what the customer asks, and if no customer order exists, the production can not start. Lastly, a hybrid policy combines the first two, which means that production could start and go until a given step without necessarily having a customer order. Still, to finish the production, we need at least a customer order.

We must define the "scale", to discuss the genericity. The scale helps define the system of interest and narrow the focus to a specific entity within the enterprise, such as a product, workstation, or work center. For example, imagine that the enterprise at study uses only a push production policy and expects to integrate the pull production policy in the future.

Suppose we want to develop a simulation model to simulate a production policy and focus on a work center as a system of interest. If the simulation model can only simulate the push production policy at the work center, it is considered specific because it can only represent the "*As-Is situation*". However, if the simulation model can simulate both the push and pull production policies at the work center, covering both the "*As-Is and To-Be situations*", it is considered partial. Finally, a model is called generic if the simulation model can simulate all production policies at the work center, including push, pull, and hybrid. In this example, the degree of genericity was determined at the work center level, the system of interest. Changing the system of interest may affect the degree of genericity of the simulation model.

In enterprise modeling, a "view" refers to a specific representation of a system or an enterprise from a particular perspective or aspect. Different views allow stakeholders to understand the system or the enterprise from different angles and can be used to support various types of analysis and decision-making.

Here are brief descriptions of some standard views in enterprise modeling frameworks:

- decision view: This view focuses on the decision-making process within the system. It identifies the decisions that must be made and the criteria used to evaluate alternatives;
- function view: This view describes the functions of the system. It identifies the primary activities and the responsibilities that the system performs to achieve its goals and objectives;
- process view: This view shows how the functions of the system are carried out through the execution of processes. It describes the sequence of steps that are performed on the components/products to achieve specific outcomes;
- data view: This view focuses on the data the system manages and uses. It identifies

the types of data needed to support the functions and processes and describes how the data is stored, processed, and transmitted within the system.

For each view, we associate a conceptual model representing a reference model of this aspect in the enterprise. Even if other authors in the literature prefer using the term perspective for this purpose, as Ulrich (2012); Whitman et al. (1999) did, we will instead use the term view for two reasons, namely:

- The perception of a specific aspect of the enterprise concerns all phases of the life cycle of development of the model, as does the usage of the view concept.
- The perception of a specific aspect of the enterprise must be consistent with the other aspects, as it is the case of the purpose of the view concept.

The instantiation of the conceptual model of a view for studying a specific aspect does not guarantee consistency with the models corresponding to the other views. The concept of view contributes to obtaining an integrated model that considers all aspects of the enterprise at study, even though each view gives rise to a different conceptual model. It also provides the possibility to modify only one specific aspect in the integrated model. Furthermore, it represents a support to generate a consistent enterprise model regarding needs, objectives, and system features.

In order to understand the importance of defining the different views, suppose that we have a conceptual model of the decision view. By instantiating the conceptual model, we can model any decision problem in the enterprise. More specifically, suppose now that we want to create an optimization model for a specific decision problem. If we design such a model and solve it, we get a solution to implement. However, we cannot predict how implementing this solution will impact the process, the data, or the performance. In this case, the purpose of the views in modeling is to provide the possibility of selecting only the dataset linked to the decision problem and modify them. After solving the problem, we can inject the solution directly into the data model and use it in the process model to assess the impacts on performance. For example, we can use a simulation model to see how the solution affects the production activity.

The consistency of the conceptual models of views such as data, process, performance, and decision helps obtain an integrated model and manages the modeling complexity.

Note that the Multi-Perspective Enterprise Modeling proposed by Ulrich (2012) was investigated, but its primary purpose is not for analyzing and designing manufacturing enterprises. Instead, it is intended for information systems and corresponding action systems, with a focus on supporting software development.

1.3.2 The Role of Enterprise Modeling

EM plays a significant role in understanding the behavior of an MS, analyzing it, in order to improving its performance (Zacharewicz et al., 2017). It provides models for decision makers to test different alternative solutions compatible with MS objectives, whether for testing new business strategies, organization modes, or processes. Furthermore, it helps create a knowledge base useful for decision support (Kosanke et al., 1999), as well as allow capitalization, since the obtained enterprise models serve as a common repository for testing forthcoming changes.

Vernadat (2014) identifies six EM principles:

1. the plural nature of the enterprise model: the enterprise model is not a single holistic model but an assembly of submodels, each depicting specific aspects;
2. the concept of modeling views (c.f. Section 1.3), used to manage the modeling complexity and to cover all essential aspects of the system. The modeling must be sure that non-overlapping views exist when choosing a set of views within the framework;
3. the three fundamental types of flows. To keep flexibility when modeling, it is fundamental to distinguish the material flow (e.g., products), the information flow (e.g., orders), and the control flow (e.g., the logic of task execution);
4. the processes versus agents. This principle considers an MS as a set of business processes executed by interacting agents, i.e., human or technical resources;
5. the business process synchronization. Three fundamental types of synchronization exist:
   - event-based synchronization, e.g., when orders generated by one process trigger other processes;
   - object-based synchronization, in which workflow steps need objects produced by one or many processes;
   - resource-based synchronization, in which the execution of a step requires the availability of some resources;
6. the concept of modeling levels, related to the concept of genericity described in Section 1.3. The author recommends using a modular and incremental modeling approach to reduce the complexity of modeling.

1.3.3 Classical Frameworks of Enterprise Modeling

We review in this section three well-known EM frameworks, which can be considered as the basis of the EM standards, namely ISO 19439 and 19440 norms (Millet et al., 2013).

1.3.3.1 GRAI Integrated Methodology

The *GRAI Integrated Methodology (GIM)* framework is based on the GRAI (Graph with Results and Actions Inter-related) model (Chen and Doumeingts, 1996). It divides the MS into three subsystems: physical (or operational), decisional and informational, and defines three dimensions, namely system views, life cycle and abstraction level. The first dimension considers functional, physical, decisional and informational views; the second deals with system phases, from the definition of the study domain to the implementation; the last refers to three levels (conceptual, structural, realizational) of system abstraction.

GIM often uses three formalisms to model an MS. IDEF0 captures both functional and physical aspects, GRAI-grid and GRAI-net handle decision-making aspects, and the Entity/Relationship model represents the informational aspect.

1.3.3.2 Computer Integrated Manufacturing Open System Architecture

The *Computer Integrated Manufacturing Open System Architecture (CIMOSA)* considers three dimensions of an enterprise. The instantiation dimension concerns the genericity degree of modeling building blocks, allowing to model generic to specific enterprises. The life cycle dimension considers the different steps of the MS life cycle from the analysis of user requirements to implementation. The view dimension partially differs from that of GIM and captures system functionality and behavior by considering function, information, resources and organisation views: by applying a life cycle from definition to implementation, we obtain the corresponding sub-models (Vernadat, 2014).

In CIMOSA, different constructs are defined for each view, and templates associated with each construct are proposed to make the modeling phase easy based on generic building blocks. For instance, to represent an enterprise activity in modeling environments, CIMOSA proposes a generalization of SADT (see Subsection 1.3.5.1). Each activity possesses three elements (function, control, resource) in its input and output.

In the following, for the sake of clarity, we will talk about the *genericity degree dimension* to refer to the instantiation dimension.

1.3.3.3 Generalised Enterprise Reference Architecture and Methodology

The *Generalised Enterprise Reference Architecture and Methodology (GERAM)* aims at covering some concepts not considered by GIM and CIMOSA. GERAM proposes the same dimensions of CIMOSA, but the life cycle dimension considers a wider set of phases as explained below:

- identification: identify business goals, stakeholders, and other relevant factors.
- concept: develop the overall concept of the enterprise system, including its purpose, scope, and high-level requirements.

- requirements: identify and document the detailed requirements of the enterprise system, including functional and non-functional requirements and any constraints.
- design: develop the detailed design of the enterprise system, including architecture, data models, and user interfaces. This may involve creating prototypes and design artifacts.
- implementation: build and test the enterprise system according to design specifications, which may involve software development, hardware installation, and component integration.
- operation: deploy and operate the enterprise system in a production environment, including ongoing maintenance, support, and monitoring.
- decommission: retire the enterprise system when it is no longer needed or is being replaced, including transferring data and other assets and disposing of equipment.

One of the major contributions of GERAM is that the view dimension becomes extensible and can be adapted to the MS object of study (Bernus et al., 2014).

1.3.4 New Frameworks of Enterprise Modeling in Industry 4.0

EM frameworks have been recently proposed to deal with more Industry 4.0-specific aspects.In the following, we shortly review the two most known examples.

1.3.4.1 Reference Architecture Modeling Industry 4.0

The *Reference Architecture Modeling Industry 4.0 (RAMI 4.0)* introduces new features as identifiability, a communication protocol (e.g., OPC UA, MQTT, HTTP), safety management to interact with advanced components (e.g., IoT), and communication modes that refer to the different ways in which devices can exchange information with each other(e.g., horizontal communication where devices are in the same layer of the RAMI 4.0 architecture, vertical communication where devices are in different layers of the RAMI 4.0 architecture). In order to represent new, complex industrial scenarios, RAMI 4.0 incorporates three types of industry 4.0 integration (c.f. Subsection 1.2.2) to address three new dimensions that decompose the MS (Pisching et al., 2018). The *vertical integration* refers to the layer dimension, a new take on the view dimension of GIM, CIMOSA, and GERAM that integrates new system layers of a smart MS (e.g., Integration, Communication, Business). *Horizontal integration* refers to the hierarchical level, or scale dimension, from product level to smart factories. Finally, *end-to-end integration* refers to the life cycle value stream dimension, which includes the value chain from product development to production and maintenance.

1.3.4.2 Industrial Internet Reference Architecture

The Industrial Internet Reference Architecture (IIRA) focuses on how smart devices and related data are handled and specifies how to address interoperability issues. Its main advantage is its open architecture. IIRA introduces four *viewpoints*, namely business, usage, functional, and implementation (Bader et al., 2019). The business viewpoint looks at the organization's goals and needs for the system, including considerations like return on investment, competitive advantage, and regulatory compliance. The IIRA usage viewpoint explains how the system will be used to meet those needs, highlighting essential parts of the system and how they work together. The functional viewpoint looks at how the system will do its job, such as processing data and staying secure. Finally, the implementation viewpoint focuses on the technical details, like the hardware and software needed to make it all work.

The viewpoints in IIRA and the views dimension already defined provide different perspectives for analyzing and designing complex systems, the viewpoints in IIRA are more focused on the specific aspects of an industrial internet system (e.g.smart manufacturing), while the views dimension is more holistic and considers the entire enterprise.

The life cycle of IIRA refers to the stages that an IIoT system goes through from conception to implementation, including design, development, testing, deployment, operation, and maintenance.

1.3.4.3 Comparison of Existing Entreprise Modeling Frameworks

Table 1.2 summarizes the dimensions of the reviewed existing EM frameworks, along with their suitability for Industry 4.0 and the "conceptual-to-implementation" feature (*Workflow* column). This feature bridges the theoretical design and practical implementation of a system, focusing on incorporating digital technologies into MSs. This feature is essential for ensuring that a system is well-conceived in theory and successfully implemented in practice. A detailed workflow outlining the implementation steps must be provided to achieve this. Often, a roadmap is required to translate the theoretical design into a functional system that meets the needs of an organization.

We distinguished the views already defined in classical frameworks and those defined in new frameworks, "smart views", because, in the context of Industry 4.0, smart views involve designing and implementing systems and processes that leverage advanced technologies to create more intelligent MSs. These smart views include concepts such as "smart assets" and "smart products", which are assets or products equipped with sensors, data processing capabilities, and connectivity to enable them to collect and share data with other systems and devices. By considering these smart views, organizations can create frameworks oriented to Industry 4.0, characterized by increased automation,

interconnectivity, and data-driven decision-making.

The life cycle of IIRA is considered "not generic" because it is focused solely on IIoT and does not cover the entire life cycle of a system. In other words, IIRA is specifically designed for IIoT systems, and its life cycle only covers the stages of development and implementation of such systems. Therefore, it does not provide a general description of a system life cycle process, including the entire lifecycle of any system from conception to retirement, such as an MS.

Table 1.2: Most relevant features of existing EM frameworks.

Framework	**Dimensions**				**I4.0-oriented**	**Workflow**
	Scale	Genericity	Views	Life cycle		
GIM	×	×	not smart	✓	×	not complete
CIMOSA	×	✓	not smart	✓	×	not complete
GERAM	×	✓	not smart	✓	×	✓
RAMI 4.0	✓	×	smart	✓	✓	×
IIRA	×	×	smart	not generic	✓	×

1.3.5 Methods and Tools of Enterprise Modeling

All EM frameworks reviewed in Subsections 1.3.3 and 1.3.4 use modeling tools and methods during the modeling process, i.e., what we called the model development life cycle phases. Those adapted for the definition phase describe the scope, processes, objectives, and requirements. This is important because they help to ensure that the model accurately reflects the needs of the enterprise and its stakeholders. In contrast, those meant for the design phase support the modeling and facilitate the implementation of models. They ensure that the model can be implemented efficiently and that they deliver value to the enterprise. For the implementation phase, methods and tools aim to provide the enterprise models such as simulation, optimization, decision, and data models by using dedicated tools to generate them. Finally, methods and tools used in the maintenance phase must allow tracking of the versioning and the update of the enterprise models, trigger changes that must occur, and keep documentation updated.

Without intending to be exhaustive, we present some well-known modeling methods and tools.

1.3.5.1 Modeling Methods

Ross (1977) introduces *SADT(structured analysis design technique)*, one of the well-known structured methods used for functional analysis to explain the details of each activity. It consists of a graphic representation in a hierarchical structure that yields a systemic view of a complex system. In EM, it is common to use SADT in the definition phase for describing MS activities with the most appropriate level of detail. An SADT is considered

a modeling method rather than a modeling tool since it uses several graphical and textual tools to represent the structure of the system at study (Kateel et al., 1996).

SADT diagram uses boxes to represent activities. The first level of system description, often called A_0, contains boxes numbered from A_1 to A_n, where n represents the number of activities in the system at study. For instance, to describe details of activity A_1, we use the same logic, i.e., boxes numbered from A_{11} to A_{1n_1}, with $1n_1$ the number of subactivities of A_1. Figure 1.1 gives an example of the SADT box structure.

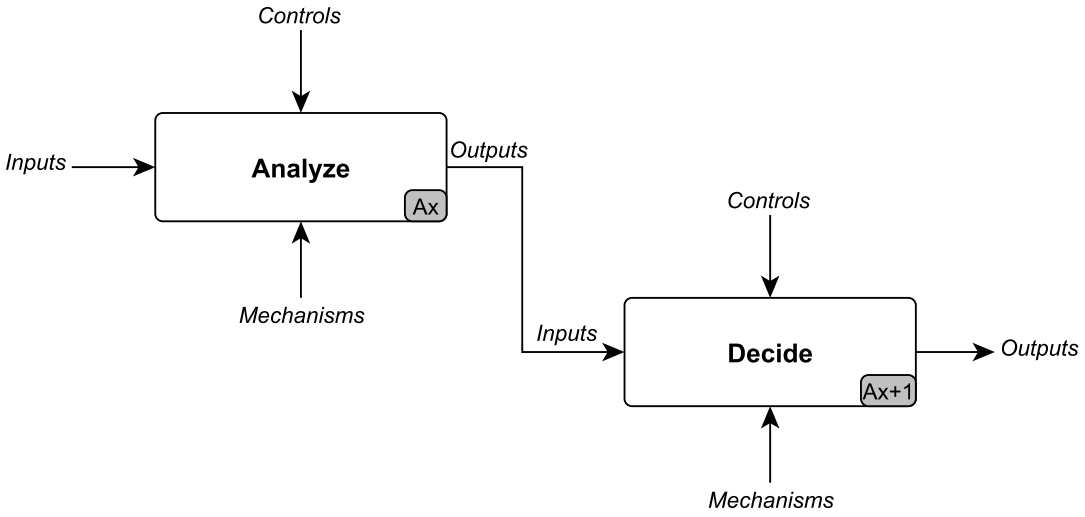

Figure 1.1: Example of SADT box structure.

Inside boxes, the infinitive verb describes the primary function of the activity. A_i refers to the activity i of the system. The inputs are material flows and data required by the activity, and controls are data events that trigger activity execution (e.g., commands or conditions). The mechanisms are the means required to accomplish the activity. Lastly, the outputs are material flow and data flow produced by the activity.

Another modeling method is *Integration DEFinition (IDEF)*. Designed during an Integrated Computer-Aided Manufacturing (ICAM) project of the US Air Force in the early 1980s, IDEF proposes a family of methods, and each one uses a diagram with a specific formalism for a given purpose (Doumeingts and Ducq, 2001). The first method, and probably the most used, is IDEF0 that uses SADT for functional modeling. Among other IDEF methods, it is worth citing IDEF1X, IDEF1, IDEF2, and IDEF3, whose objectives are Data modeling, Information Modeling, Simulation model design, and Process description capture, respectively. The reader interested in details about IDEF methods could consult Mayer et al. (1994), where basic concepts and modeling guidelines of the IDEF methods are described.

The *object-oriented modeling method* is used in the design phase of model development. As this method is programming-oriented, it may considerably shorten the development

life cycle. Based on the principles of object-oriented programming, it translates each MS component into an equivalent element of code, i.e., an object. In EM, and especially for simulation modeling environments, the object-oriented modeling technique provides high flexibility to quickly generate models and modify their features (Anglani et al., 2002).

1.3.5.2 Modeling Tools

As defined in Subsection 1.3.1, the modeling tools are techniques that use diagrams to represent concepts and are exploited in the modeling methods. The object management group (OMG) proposes *UML (Unified Modelling Language)*, a language to support the object-oriented modeling method. It proposes different diagrams to study different aspects of complex systems, such as class diagrams, sequence diagrams, and case diagrams. Figure 1.2 gives an example of a class diagram, a static diagram commonly used in conceptual modeling to describe the structure of system. The concept of class is an abstraction to represent any object with its properties and behavior. It is one of the basic elements, for instance, of object-oriented programming.

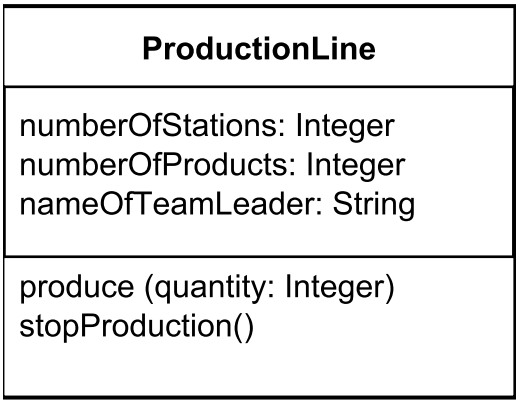

Figure 1.2: Example of class diagram.

As we can see, the class representation contains three parts. The uppermost part contains the class name (ProductionLine in the example of Figure 1.2). The center part contains attributes that reflect the class features (numberOfProducts in the Figure 1.2). Lastly, the bottom part includes methods of the class produce(quantity) in the Figure 1.2, which are the possible operations that an object of the class can make. Attributes and methods can have different degrees of visibility for other objects, depending on the class of these latter.

Different relationships may exist among classes. In UML, relationships define how two or more elements can relate to each other at runtime. The four types of UML relationships are *Association*, *Dependency*, *Inheritance*, and *Realization*. Each type of relationship serves a different purpose in modeling the system. An association is a structural relationship that signifies that objects within a system can be connected or

associated with other objects. Constraints can be applied to the association relationship to specify further the nature of the connection between the associated objects. For instance, UML uses a notation called multiplicity to precise the number of objects of each class participating. Figure 1.3 defines the Five types of associations.

- Bi-directional means that the classes reference each other. Example of this association is given in Figure 1.3, an Employee can have multiple Skills, and a Skill can belong to multiple Employees.
- Uni-directional indicates that only one class references the other class. Example of a Customer that places one or several Orders, and each Order belongs to a single Customer. Using this association relationship, any modifications made to the Customer class would not affect the Order class since the association does not reference the Customer from the Order.
- An aggregation is a binary association, i.e., it could not include more than two classes as a classical association does. This relationship represents the case when class A is a part of class B. For instance, a Wheel class is a part of a Car class. Furthermore, a Wheel can exist independently of a Car.
- A composition is also a binary association. However, the relationship represents the case when class A is a whole part of class B. A classic example of composition is an association between Folder and File classes. The File is a whole part of a Folder which means that the Folder may contain many files, but each File has exactly one Folder parent. Furthermore, the delete action of the Folder automatically removes all files contained in it.
- Reflexive is an association to depict that links could exist among objects of the same class. For example, the car assembly process involves several predecessor processes, such as creating the engine, the frame, and the interior components. Once these processes are completed, the assembly process can begin. After the assembly, the car may undergo further processes, such as painting, quality control, and shipping. These processes are considered the successors of the assembly process. Therefore, the car assembly process can be seen as a process with several predecessors and successor processes.

A dependency relationship describes how different elements within a system rely on each other. It is a unidirectional relationship indicating that a change in one element may affect another.

An inheritance relationship describes that one class is a subclass of the other class (specialization); or inversely, one class is a superclass of the other (generalization). In an inheritance relationship, one entity is a parent, and another is said to be a child.

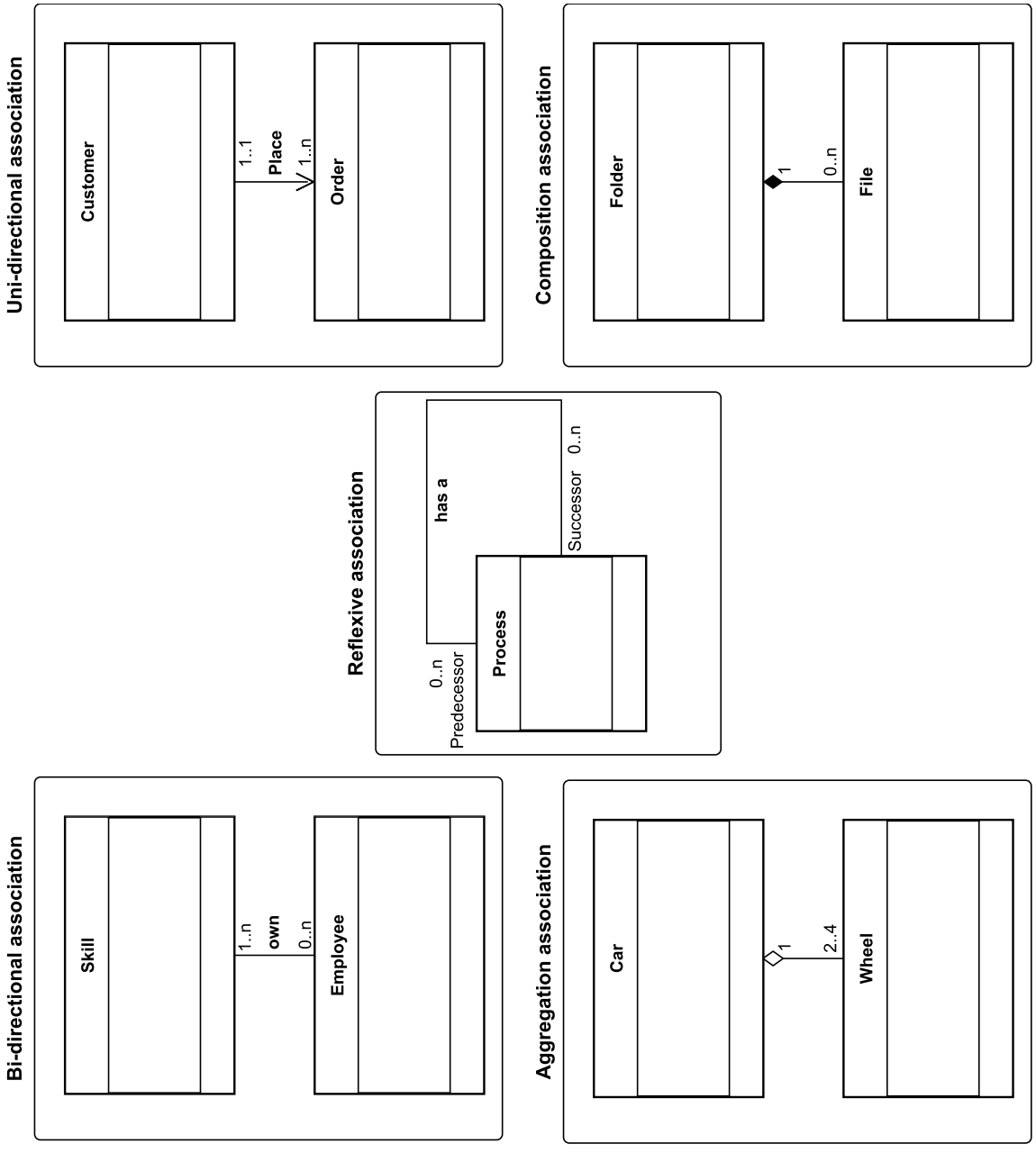

Figure 1.3: Examples of the association types in UML.

In UML, a realization relationship defines a relationship between two entities where one entity represents a responsibility that is not implemented by itself, and the other entity implements it. This relationship is commonly found in the case of interfaces. In UML, an interface is a type of classifier that defines a set of operations or methods that can be implemented by classes or other interfaces that implement the interface. It specifies a contract that must be followed by any class or interface that implements it. When an entity, such as a class or interface, implements an interface, it is required to provide an implementation for all the methods defined in the interface. The implementation of these methods is what realizes the responsibilities defined by the interface.

We briefly reviewed the basic concepts of the class diagram. However, for more details about UML, the ISO 19501 norm ISO (2005) provides semantics, notations, and diagrams.

Systems Modeling Language (SysML) is a language that keeps just a part of the UML diagrams and adds two specific diagrams, introduced to better respond to the system engineering needs. Nevertheless, in what concerns EM, Vernadat (2002) proposes to use a *Unified Enterprise Modeling Language (UEML)* as a specialized language for enterprise modeling purposes. UEML aims to have a uniform interface and resorts to UML to define its core constructs. However, it is worth noting that UEML is not as widely used or recognized as UML, which remains the dominant modeling language for various applications, including software engineering and systems modeling.

UML is still the preferred modeling tool for many practitioners in the field of EM (Enterprise Modeling), as evidenced by its continued use in various works related to EM. This preference is further supported by the fact that the latest version of the ISO 19740 standard, which deals with EM, specifically references and utilizes UML rather than UEML or SysML.

Even though EM and Decision Support Systems are strongly related, few works highlight this relation explicitly. According to Zolghadri et al. (2001), the first logical step to designing Decision Support Systems is to analyze the decision system. For this purpose, EM methodologies are suitable since they propose to take into account other aspects of an MS while focusing on the decision aspect. When combined, the models obtained from the modeling tools can help decision-makers build powerful Decision Support Systems for designing or redesigning MSs. However, to take advantage of EM, the design of Decision Support Systems must respect the EM principles, which allows to likely obtain a Decision Support System consistent with the company guidelines and requirements. In the following, we dedicate Section 1.4 to going deep with the decision making by focusing on the decision-support in manufacturing systems.

1.4 Decision Making and Decision-Support in Manufacturing Systems

Mintzberg et al. (1976) considers *a decision* as "*a specific commitment to action*". Most of the time in MSs, the need of a decision rises after the detection of a problem. For instance, when confronted with a quality problem in production, managers may decide to analyze e.g., the capability of machines, the skills of operators, and or the quality of components delivered by suppliers.

In this thesis, *a decision problem* consists of determining how a decision-maker can use some MS resources to reach some economic or performance objectives. The literature describes several decision problems. An example is the capacity investment problem, which determines the required capacities over the long term and specifies where and when to introduce the product to optimize the capacity invested (Martínez-Costa et al., 2014). Another example of problem from the supply activity is the choice of the line feeding policy, which consists to specify the rules (e.g., the stock level, the component type) used for delivering the components from the warehouse or the supermarkets to the workstations. Finally, the batching problem is an example from the storage activity, which refers to combining multiple orders for picking into a single tour or picking list. The objective of batching is to increase the efficiency of the order picking process and reduce the total travel time of the pickers.

Turban et al. (2005) defines *decision making* as a process of choice among several solutions and actions to achieve one or more specific goals or objectives. This process starts with scanning the environment, collecting information, and defining the problem. Afterward, it continues by building the model, searching, and developing alternatives for solving the problem, and assessing the outcomes of utilizing these alternatives. Finally, it ends by choosing one of the alternatives. In the literature, several frameworks exist to organize the decision making, which involves information and people and several phases depending on the used framework (Mintzberg et al., 1976). For instance, Simon is a well-known framework that uses three phases known as an *Intelligence-Design-Choice (IDC)* trichotomy to summarize the decision making process (Simon, 1960). The design phase consists in building a model.

For each problem, a model is created and several use cases can be derived for, which present instances of the problem. An *instance* of a model means a model generated for the use case from the model of the problem. The *instantiation* of the model means the creation of an instance of the model.

Three modes of decision making exist, descriptive, normative, and prescriptive. As Dillon (1998) explains, the descriptive mode focuses on what decision-makers actually do. In contrast, the normative mode focuses on what decision-makers should do in theory. The prescriptive mode examines what decision-makers should and can do. The

decision making model depends on the mode. Operations Research (OR) and Decision Sciences are disciplines meant for developing normative models (e.g., analytical performance models, c.f. Subsection 1.4.3.1) and prescriptive models (e.g., optimization models), while Behavioral Decision Theory is the reference of descriptive models (Weber and Coskunoglu, 1990).

We divide the decision-making architecture of a company into three levels, namely strategic, tactical and operational, which are performed by top, middle and supervisory management, respectively (Er, 1988). The *time horizon* of a decision is the period affected by the decision making. It distinguishes strategic, tactical, and operational decisions, i.e., long-, medium-, and short-term decisions. Time horizon impacts both the level of uncertainty and the level of abstraction of the MS description, ultimately determining the type of model used to support the decision. Depending on the industrial sector of a company, strategic decisions (rather than tactical or operational) will be associated with different timespans. For instance, for the automotive sector, the timespan affected by strategic decisions can vary from two to five years, while for the aeronautical sector is from ten to thirty years.

Er (1988) distinguishes three types of problems, structured, semi-structured, and unstructured. Structured problems are repetitive routines with well-known procedures to find solutions. Unstructured problems are fuzzy and no standard solution methods exist, therefore, human judgment and intuition are the references for decision making. The semi-structured problems are those for which some phases of the decision-making process are structured, and other phases are not. They benefit calculator capability but require human judgment.

Decision support consists of using one or several modes to conduct the decision-making process, i.e., formulating the problem, developing the corresponding model, and making a choice. When computers are used to perform the decision-making process, we talk about decision support systems (Keen, 1980). The term *Decision support systems (DSSs)* was coined by Keen and Morton (1978) to indicate computer systems that impact decisions and act as a supportive tool, although human judgment is still required, despite the computer and analytical assistance. Keen (1980) considers DSS as an interactive computer-based system that help decision-makers use data and models to solve unstructured problems. In contrast, Turban et al. (2005) state that DSSs serve for semi-structured problems. Nowadays, we use DSS to tackle even structured problems since knowledge could transform unstructured problems into structured ones.

Nowadays, in broader sense, the term DSS also defines an area of the Information Systems (ISs) discipline that supports and improves managerial decision making (Arnott and Pervan, 2014).

Typically, the architecture of a DSS includes three components: the database management system to access and manipulate the existing data stored in databases; the

model management system, that uses models to represent reality and analyze data; The user interface, that provides support for the interaction of the user with the system.

1.4.1 Importance of Decision-Support Systems in Manufacturing Systems

Decision problems in MSs are complex due to many factors, among which are the multiple conflicting criteria and uncertainty.

Despite these factors, decision makers at different levels of the decision-making architecture need to make the best possible decision. This is when DSS come into play.

Arnott and Dodson (2008) state that using DSSs for decision making has the potential to alter the nature of an organization significantly. First and foremost, DSS drives decision makers to understand the decision problem better and with less effort. The aim is to support decision makers in the decision-making process, not replace their actions as expert systems do (Pick, 2008). Furthermore, DSSs reduce the time and cost of decisions by using capability of calculators to explore several alternatives. Likewise, they improve the efficiency of the decision process by using models and data instead of intuition, therefore increasing confidence in the decisions. Another great advantage of DSSs is their accessibility to non-expert users. Decision makers can effectively manipulate complex models and focus rather on the performance assessment (Eom, 1999). Since a DSS is a computer-based solution, it enables better communication among the members concerned by decisions, who can then share the same vision.

Based on these benefits, the importance of DSSs seems evident since they provide powerful computation, high-quality support, easy data access, and improved communication. Along with ICTs evolution, DSSs allow to face the constant change of source of information and the requirement of making decisions quickly.

1.4.2 Types of Decision Support Systems

We classify DSS types according to two criteria namely, functions and applications.

Based on the function of DSSs, Felsberger et al. (2016) evokes three DSSs types. Passive DSSs support the decision-making process but do not suggest decision solutions, as active DSSs do. Cooperative DSSs propose solutions and involve decision makers in adapting decisions and reinjecting them into the system for validation.

In terms of the applications, we distinguish five DSS types. Model-driven DSSs are systems that leverage models to assist decision-makers in making informed decisions. For instance, they provide decision support through the use of analytic, optimization, and simulation models. Data-driven DSSs allow access and manipulation of well-structured data, like Business Intelligence (BI) systems, so as to assist in decision-making by triggering and analyzing data stored in databases.

Communication-driven DSSs are collaborative decision systems that employ communication technologies to share information and connect decision makers. They aim to create a group of decision makers and find solutions collaboratively. Document-driven DSSs collect information from any document (e.g., images, videos, hypertext, sounds) and exploit tools to transform them into a standardized format for decision-making support. Lastly, Knowledge-driven DSSs use technologies such as AI to identify specific knowledge and exploit it in the decision-making process.

Either qualitative or quantitative modeling techniques can be used to deal with decision problems. Qualitative modeling represents different aspects of systems in a symbolic and human-like manner (Forbus, 2008). However, to solve structured problems, we mostly use quantitative modeling. For instance, OR exploits mathematical modeling and uses algebraic expressions to describe problems. In the following, we propose to detail the model-driven DSSs, and more specifically, DSSs based on the quantitative modeling.

1.4.3 Quantitative Modeling

From the perspective of quantitative modeling, Biswas and Narahari (2004) distinguish three model types: analytical performance, optimization, and simulation models. Additionally, combining these models can enhance decision-making abilities. As a result, we suggest examining each model type in the following sections.

1.4.3.1 Analytical Performance Models

Analytic performance models are high abstraction models such as Markov models, Petri nets, and queuing models (Biswas and Narahari, 2004). Markov models are meant for systems with state-dependent behavior, i.e., the next state of the system depends on the sequence of the previous states (Garrido, 2002). The system has each time a probability to transit from one state to another one. Queuing models are a type of Markov models used to represent behavior with a large state space. Petri nets model the behavior of complex systems using a graphical tool with a specific formalism (Zhou and DiCesare, 1993). Nodes represent the states of the system, and the notion of token indicates the conditions and events wihtin the system (Reisig, 1985).

Using these analytical models to determine the performance of larger systems such as MSs can become a complex task due to nonlinear behavior and uncertainties, as well as the necessity of making assumptions for parameter estimations such as processing times, failure rates, or resource capacities. Therefore, other types of models are necessary, such as optimization models.

1.4.3.2 Optimization Models

Optimization models are mathematical models that seek to minimize or maximize one or several criteria or objective functions. As described by Turban et al. (2005), the decision-making process when using optimization models differs from intelligence-design-choice (IDC) (see Section 1.4) and follows five phases instead:

1. The definition of the problem;
2. The classification of the problem into a reference category;
3. The building of an optimization model;
4. The search and evaluation of the problem solutions;
5. The choice and recommendation of solutions.

Usually, the modeling of decision problems involves taking into account the constraints imposed by the behavior of the MS. Building the mathematical model requires defining decision variables and using them to represent the decision alternatives and to express the objective function(s) and constraints of the problem in the form of mathematical relations over the decision variables. The type of decision variables (e.g., integers, real), their number, and the nature of mathematical relations (e.g., linear, quadratic) define the complexity of the optimization problem. Depending on it, one chooses a most suitable optimization method to explore the search space and find solutions. Exact methods ensure optimal solutions; heuristic methods obtain near-optimal solutions in a reasonable computational time without providing optimality proof; approximate methods yield sub-optimal solution with a guarantee in terms of distance from optimality.

One of the main advantages of optimization is quantitative decision support. It provides optimal or near-optimal solutions for several decision problems, thus allowing cost-saving. When modeling the complexity of reality becomes difficult or impossible by optimization models, one common alternative, among others, is resorting to the simulation models.

1.4.3.3 Simulation Models

Simulation aims to imitate reality based on a model of the real system, conduct experiments to understand the system behavior, and evaluate several operating strategies (Shannon, 1998). Negahban and Smith (2014) recognize the role of simulation in evaluating the performance of MSs at different levels, from design to operational level. According to Turban et al. (2011), simulation models are descriptive models that aim to represent things "*as they are or as they are believed to be*". They use a computer program to study the behavior of the modeled system and answer some "*what-if*" questions.

System modeling could be either static or dynamic. In static modeling, we do not consider the time factor. The obtained model represents only the state of the system at a specific moment (e.g., Monte-Carlo simulation). In contrast, dynamic modeling considers the time factor, and the model can simulate the behavior of the system over a time horizon. Most of the time, in MSs, the simulation is used for the evaluation of systems with complex dynamics. Depending on whether the model integrates uncertainty or not, we talk about stochastic or deterministic model, respectively.

The review of Jahangirian et al. (2010) shows that among the most used simulation methods are:

- System dynamics (SDs) is based on dynamic modeling with a high abstraction. Time is represented continuously rather than discretely. It considers that the state of a system changes constantly. The system is modeled using a series of mathematical equations that describe the relationships between the different components of the system. This technique is adapted to represent the fluid dynamics, chemical reactions, and trajectories of a vehicle on a road, among others.
- Discrete-event simulation (DES) is one of the most common decision support techniques to understand the dynamics of MSs (Negahban and Smith, 2014). It is based dynamic modeling but with a low level of abstraction. The system being simulated is defined as a set of entities that interact with each other in different ways, and a set of events that can occur to change the system state. These entities and events are modeled as objects in the simulation. The simulation progresses through time in small steps, where each step represents a discrete point in time. At each step, the simulation checks the system state, identifies possible events, and then selects the next event to occur. The selected event then updates the system state based on its known effect on system behavior. The timing of events is random, but their effects are specified during the modeling process.
- Similarly, agent-based simulation (ABS) is a low-abstraction method that describes a system from the perspective of individual objects or autonomous entities (so-called agents) interacting with each other and with the environment (Borshchev, 2014). Each agent has its own set of rules, behaviors, and decision-making processes that govern its actions and interactions with other agents and the environment. The simulation typically proceeds through discrete time steps, during which agents perform their actions and interact with their surroundings. The results of each time step are then aggregated to produce a comprehensive understanding of how the whole system evolves over time. In ABS, agents can be anything from a resource in an MSs, to people in a social network, to vehicles on a roadway.

We talk about a hybrid simulation when we use more than one simulation method (Jahangirian et al., 2010).

Brailsford et al. (2019) describe the main phases of the process of simulation modeling. First of all, we define the problem and select the simulation method. The building of the conceptual model, the most important phase in a simulation project, follows. Robinson (2008) defines it as a process of abstracting a model from a real system and considers it an art rather than a science. Then, the computer implementation phase consists of creating a computer program of the conceptual model. Data collection is then required to run the program. The model parameterization phase (e.g., number and length of replications, execution time, stop conditions) allows specifying simulation conditions. Next, the verification and validation (V&V) phase requires a good understanding of the system under study and generally the point of view of experts. Verification ensures that the computer program is running without errors, and that the model behavior corresponds to the intended conceptual model. In comparison, validation aims to check whether the model behavior produced by the simulation model represents the reality adequately or not (Sargent, 2013). In this phase, it is recommended to use key performance indicators (KPIs) to quickly judge the model behavior (Saetta et al., 2012). Nevertheless, even after the validation phase, it is mandatory to generate a set of scenarios to be tested, especially in stochastic models (Ingalls, 2001; Terzi and Cavalieri, 2004). Besides, the design of experiments (DOEs) introduced by Fisher (1935) use the same idea based on statistical techniques to create a set of experiments. DOE consists in testing several scenarios corresponding to various combinations of input parameters, called *factors*, to find the most influential, based on the model outputs, called *responses*. Another objective of DOEs is to determine interactions exist among factors. Lastly, the analysis and presentation of results concern the interpretation of the model's outputs to assist managers in decision-making in the real world.

We summarize the main advantages of simulation in three points:

- anticipation of decisions to avoid design, planning, or control problems;
- decision-making with hindsight using a virtual evaluation;
- shortened development cycles.

Likewise, we summarize the main disadvantages of the simulation in three points:

- the modeling process to generate a simulation model is a difficult task asking for the expertise and a long time;
- the accuracy of the simulation depends on the accuracy of the input data and assumptions made during the modeling process;
- the simulation can be computationally intensive, requiring significant computational resources to simulate large-scale systems.

In what follows, the terms *optimization-based DSS* or *simulation-based DSS* will indicate that the model-driven DSS use *optimization models* or *simulation models*, respectively.

1.4.4 Decision Making and Decision Support Systems in the Industry 4.0

Over the last decades, Information and Communication Technologies (ICTs) have become an essential source of information for decision making. This is even true with Industry 4.0 as nowadays ICTs can access huge amounts of data and valuable knowledge. However, decision makers must be insightful and make decisions in shorter time. Moreover, such huge amount of data must be exploited correctly to guarantee a better quality of decision making. In the following, we briefly identify decision-making evolutions in Industry 4.0, then discuss ongoing changes on DSSs.

Based on our comparison of MSs types (c.f. Subsection 1.2.1.2), although RMSs can handle substantial modifications in product design, their centralized control logic is not fitting the requirements of the Industry 4.0 context. A centralized control approach may be adopted in the case of model-driven DSS for efficient coordination over decision-making processes and facilitating the integration of different data sources. However, in the case of data-driven DSS, a *decentralized control approach* with a coordination mechanism is more adapted to handle the huge quantities of data involved in making decisions. Furthermore, this approach is also compatible with the connectivity objective of Industry 4.0 discussed in Section 1.1. The emergence of smart devices and of the M2M (machine-to-machine) principle, along with the development of the CC (cloud computing) technology, are additional reasons to use such an approach for decision making in Industry 4.0. Typically, in practice, a combination of the two may be required, i.e., adopting the decentralized control approach among entities, but having a supervision level.

The challenges of MSs within Industry 4.0, such as the short PLC and the integration of data (c.f. Subsection 1.2.2), impact decisions temporality, and, as a consequence, the time horizon of decisions. For instance, some decisions that previously were made on a long-term basis, and were hence considered strategic decisions, can now be dealt with a shorter term, thus becoming tactical, and sometimes even operational, decision. At the same time, new paradigms are emerging, such as real-time decision making.

Moreover, technology evolution within Industry 4.0 (c.f. Subsection 1.2.3) has also contributed to the development of modern concepts in MSs, such as CPS. As new MSs differ from the ancient types w.r.t. the modeling and behavior, classical KPIs are no longer sufficient to assess their performance. Consequently, new KPIs oriented towards Industry 4.0 should emerge (e.g., ergonomics, agility). Ergonomics is one aspect covered by the human-centric approach evoked as one of the three pillars of Industry 5.0. For the same reasons explained before in Subsection 1.2.2, this manuscript will consider it involved in the Industry 4.0 context.

To meet these decision-making evolutions and sustain competitive advantage, industrial companies are looking for DSSs that are adapted to the Industry 4.0 environment. Powell

(2019) identifies that modern DSSs are enhanced with the possibility to evaluate the output of optimization models with simulation. Hence, an integrated approach with the coupling between optimization and simulation models is worth considering since decision making will benefit from both optimization and simulation advantages.

In the literature, four strategies exist to combine simulation and optimization (Ladier et al., 2014). Table 1.3 groups these strategies into post-treatment and integrated approaches:

Table 1.3: Possible strategies for combining simulation and optimization models.

Approaches	**Strategies**	
Post-treatment	$Optimisation \rightarrow Simulation(S1)$	$Simulation \rightarrow Optimisation(S2)$
Integrated	$Optimisation \subset Simulation(S3)$	$Simulation \subset Optimisation(S4)$

In strategy S1, the optimization results are included in the simulation model as inputs in order to be simulated and their performance assessed. Inversely, strategy S2 uses simulation to generate the data required for the configuration of the optimization model. In these approaches, optimization and simulation are used in sequence: however, it is possible to set up a feedback loop in both cases, in order to increase efficiency.

In the integrated strategy S3, the simulation launches optimization and uses its results while running, as an evaluation tool. Optimization provides decisions, i.e., values to inject to continue running the simulation model. In strategy S4, the simulation model is evaluated during the optimization process, then simulation results guide the optimization method in exploring search space.

When both simulation and optimization models are used in model-driven DSSs, the term *Optimization/Simulation-Based DSS* is used. Nevertheless, for complex systems like MSs, conceptual frameworks to create such DSS are needed.

1.5 Open Research Questions

Figures 1.4, 1.5 and 1.6 structure and summarize all concepts presented in the Chapter 1.

First, Figure 1.4 provides a comprehensive summary of the key concepts of the Industry 4.0 revolution. It covers various integration types and highlights the seven challenges discussed in Subsection 1.2.2. Additionally, it classifies the technologies of Industry 4.0 presented in Subsection 1.2.3.

In Figure 1.5 summarizes the EM discipline and refers to the norm for defining basic concepts and core constructs. It highlights the EM principles suggested in the literature and introduces concepts related to the EM methodology. These concepts are consolidated into a structured approach that guides the modeling process, EM frameworks based on dimensions, an enterprise architecture incorporating reference models, and commonly

used EM methods and tools in EM methodologies. Finally, a link between EM and DSS is illustrated to show that EMs are used for building DSSs.

Lastly, Figure 1.6, an overview of decision-making in MSs is presented. It outlines the modes and time-horizons of decision making in MSs presented in Section 1.4. Also, it details the decision-making process based on DSSs as described in Subsection 1.4.1 and offers the first insights into the evolution of decision-making in Industry 4.0, which are discussed in Subsection 1.4.4.

The investigation conducted in the Chapter 1 allowed us to raise the following research questions.

Research Question 1

> What are the existing decision problems and their interactions in Manufacturing Systems? How can we model these decision interactions in Manufacturing Systems?

In the context of an MS, the creation of decision-support systems supposes a deep understanding of the decisional aspect of MSs, i.e., of the decision problems involved. Suggesting models for tackling a specific decision problem might help decision makers choose among available alternatives. However, decision makers will not be able to explain how this decision will impact other decisions and how MS performance may change. Provide practitioners with robust and integrated models to achieve the overall MS performance required, based on a holistic view of the structure of decision problems and how they interact, is a highly complex task, due to the complexity of MSs. The challenge is to systematically review all the decisions involved in an MS and investigate their interactions. The ultimate objective is to provide decision makers a deep understanding of the impact of their choices. As far as we know, no work exists that considers all MS decisions as a whole and studies their interactions. The main contribution of Chapter 3 is to try to fill this gap by providing a complete taxonomy and cartography of decision problems related to MS activities (production, supply, and storage) and then study their interactions.

Research Question 2

> What is the impact of the advent of Industry 4.0 on decision making in Manufatcuring Systems?

Designing a DSS adapted to the Industry 4.0 technical and technological context raises the challenge to assess the evolution of decision making in manufacturing systems in a broad sense, i.e., with the same viewpoint of Section 1.5.

To the best of our knowledge, the scientific literature lacks research works that offer a holistic investigation on how decision making is changing within Industry 4.0. The main contribution of Chapter 2 is to study the decision making evolution within Industry 4.0. The framework designed in Chapter 3 will serve as a tool to illustrate this evolution.

Research Question 3

> What new features should an Enteprise Modeling Framework incorporate to allow dealing with the decision making changes identified in Industry 4.0 and derive Industry 4.0-oriented Decision-Support Systems?

The evolution of industrial paradigms imposes the evolution of DSS towards Industry 4.0-oriented DSSs, which should include the uncertain aspect and be redesigned and implemented in parallel to the MS, if not in advance (Felsberger et al., 2016). However, such more advanced DSSs cannot be designed without rethinking Enterprise Modeling methodologies.

The more recent EM frameworks, despite integrating at the conceptual level the specific concepts of Industry 4.0, are not yet able to carry out the modeling up to the implementation of an agile, quickly reconfigurable Optimization/Simulation-based DSS.

The objective of Chapter 4 is to propose a new Enterprise Modeling Framework called MEMO I4.0, which features this conceptual-to-implementation capability. The chapter will also focus on how MEMO I4.0 can assist to generate an MS model, quickly scale it up or reconfigure it whenever a new MS configuration is at study.

To show a real application of MEMO I4.0 and its efficiency to derive Industry 4.0-oriented DSS, Chapter 5 will focus on case studies taken from the reality of elm.leblanc company. We will discuss the results obtained, and we will recommend guidelines to managers for implementing the proposed solutions in the real world.

In Section 5.4, We will summarize all contributions of the manuscript and provide some perspectives on our research works. First, we will propose generalizing the approach for investigating decision problems in manufacturing systems (MSs) to the supply chain. Then, once standardization becomes possible, we will ask to update MS features for Industry 4.0 and create semantic models or ontologies to enhance the proposed metamodels. We will specify that the improvement of interoperability within Industry 4.0 can be achieved with a unified namespace (UNS) and using an administration shell for modularity. Regarding the view dimension extension, we will suggest incorporating a business view in MEMO I4.0. We will also recommend researching tools, languages, and ontologies compliant with the MEMO I4.0 framework and trying to apply it to different domains. Furthermore, we will propose investigating ML techniques and thoroughly studying the implications of integrating IT and OT systems. Finally, we will highlight the necessity of finding the network infrastructure and protocols to handle data generated by cyber-physical systems and integrating emerging technologies into Industry 4.0.

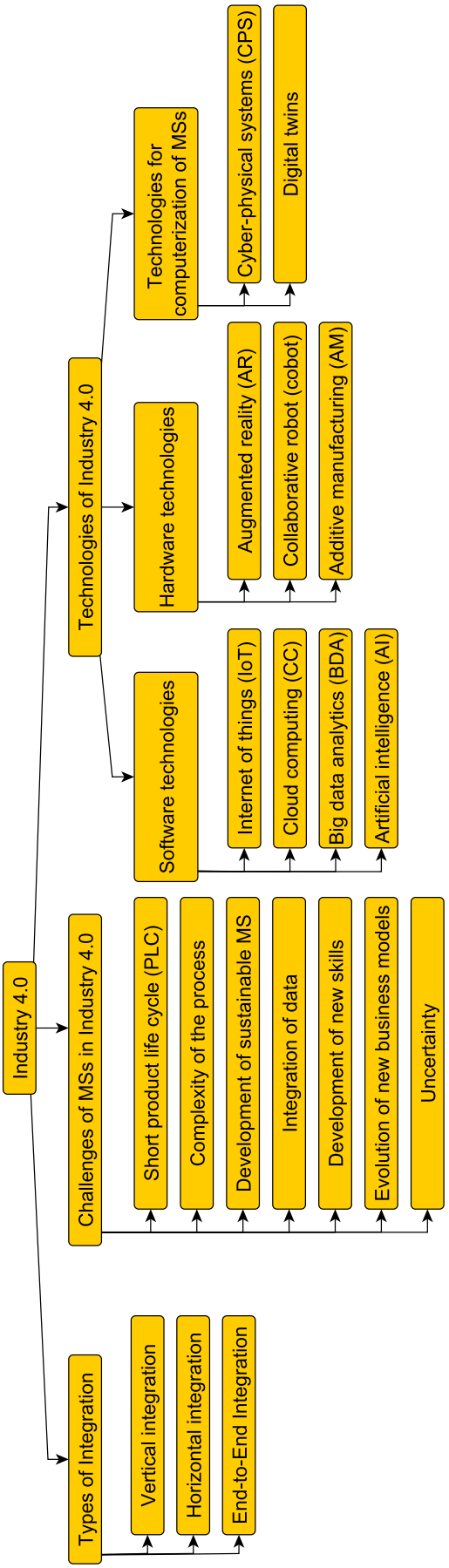

Figure 1.4: Summary of manufacturing systems (for details see Subsections 1.2.2 and 1.2.3).

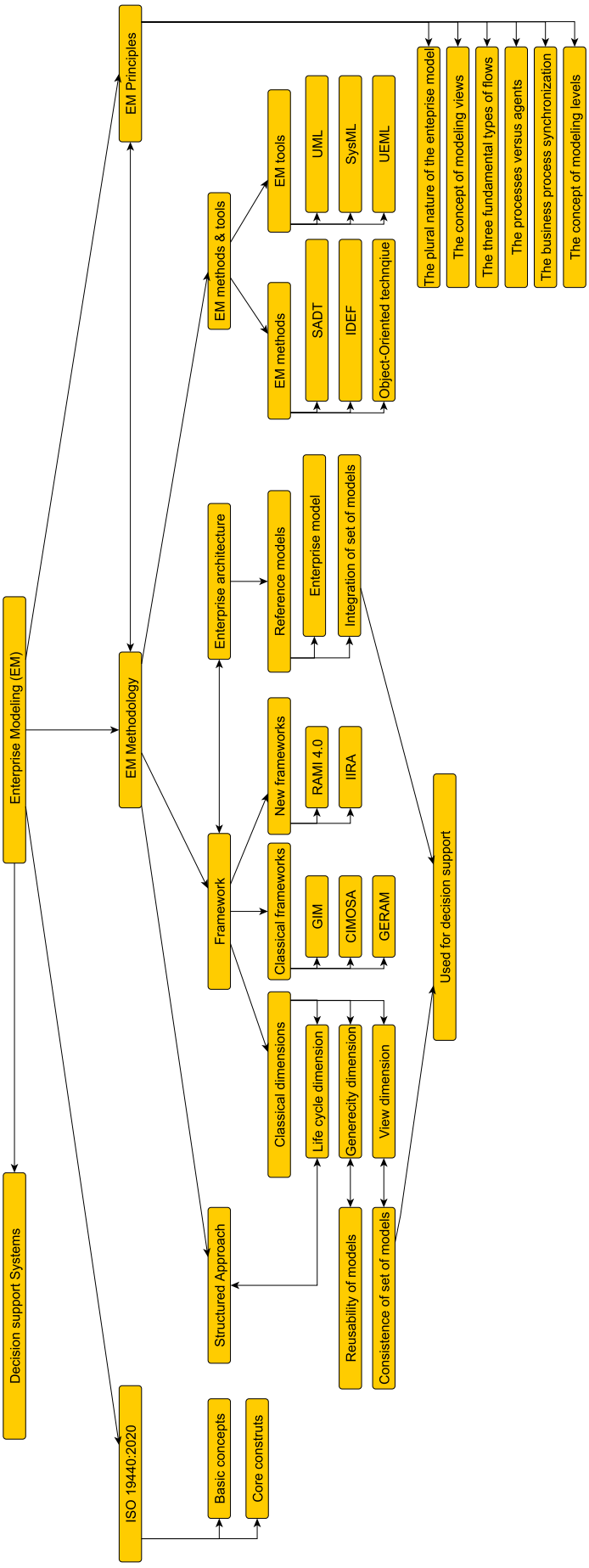

Figure 1.5: Summary of Enterprise Modeling.

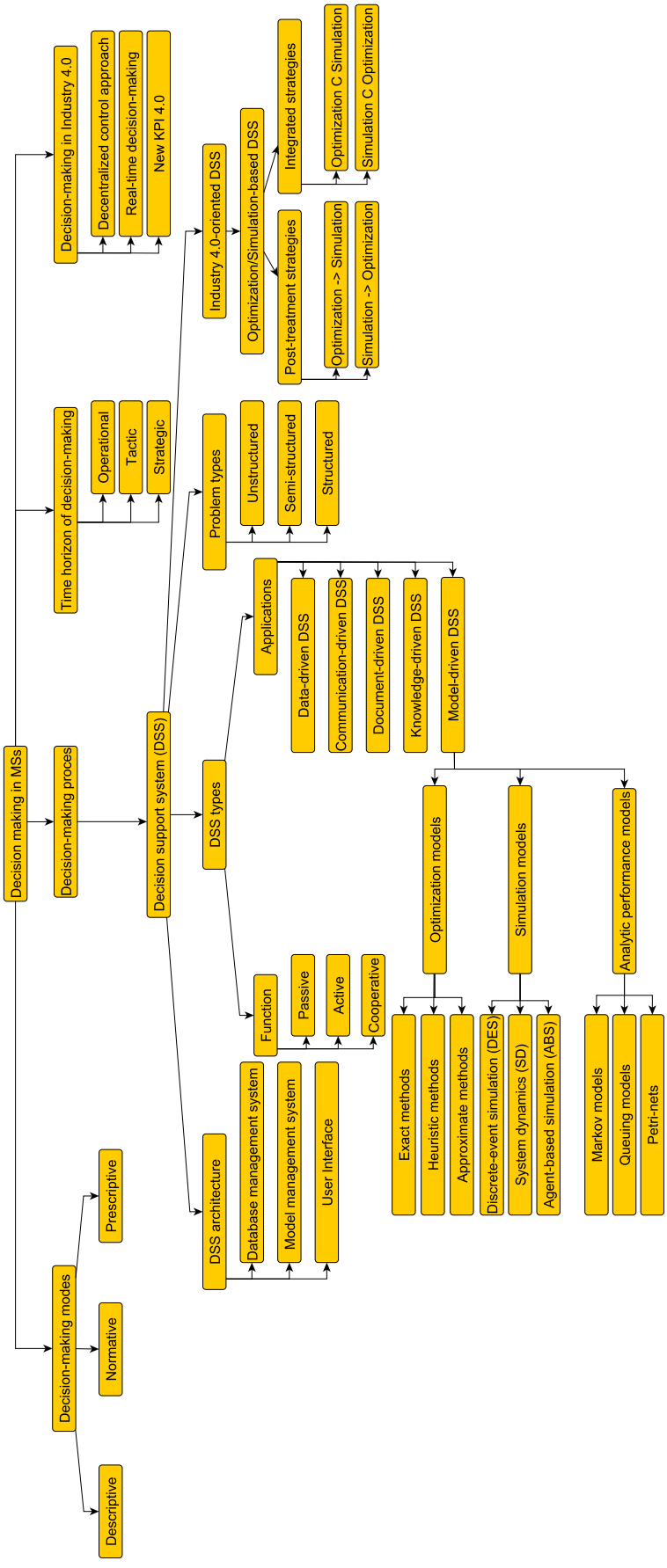

Figure 1.6: Summary of concepts related to the decision making in Manufacturing Systems.

Chapter 2

Decision Problems in Manufacturing Systems and their Evolutions in the Industry 4.0

Résumé

Ce chapitre aborde les problèmes de prise de décision dans les systèmes de production et leur évolution à l'ère de l'Industrie 4.0. Il commence par identifier les problèmes de décision existants dans chaque activité principale des systèmes de production, à savoir: la production, l'approvisionnement et le stockage. Ensuite, il met en évidence les principaux leviers de transformation de l'ancien scénario vers le scénario Industrie 4.0. Cette transformation entraîne un changement de perception de l'entreprise dans l'Industrie 4.0, nécessitant ainsi une nouvelle approche de la modélisation d'entreprise. Par la suite, il analyse l'impact des leviers de transformation sur la prise de décision pour définir les exigences de l'Industrie 4.0. Également, il examine l'évolution des indicateurs clés de performances des systèmes de production dans l'Industrie 4.0, en mettant en évidence les changements fondamentaux dans la conception et la gestion des systèmes de production. Ce chapitre aborde ensuite les évolutions de la prise de décision dans l'Industrie 4.0 en se focalisant sur l'horizon temporel des décisions, et le type de système de production. En conclusion, le chapitre identifie les défis de la prise de décision à l'ère de l'Industrie 4.0, et propose des leviers de transformation pour mener la transition vers l'Industrie 4.0. Enfin, il suggère de développer une taxonomie des décisions prises dans les systèmes de production et de concevoir de nouveaux outils favorisant le développement de modèles intégrés. Le but ultime est de créer des systèmes d'aide à la décision adaptés à l'Industrie 4.0.

Contents

2.1 Introduction 58
2.2 Decision Problems in Manufacturing Systems 59
2.2.1 Decisions related to Production Activity 59
2.2.2 Decisions related to the Supply Activity 64
2.2.3 Decisions related to the Storage Activity 66
2.3 Transformation of the Old Industrial Scenario into Industry 4.0 Scenario 71
2.3.1 Technical Evolution 74
2.3.2 Organizational Evolution 77
2.3.3 Transversal Evolution 81
2.4 Evolution of Decision Problems in Manufacturing Systems in Industry 4.0 87
2.4.1 Evolution of Time Horizon of Decision Problems in MS 87
2.4.2 Evolution of Manufacturing Systems Type 93
2.5 Conclusion 93

2.1 Introduction

Decision makers in Manufacturing Systems (MSs)(e.g., directors, managers, engineers) must face decision problems concerning different MS activities and different time horizons. These problems may concern, for example, the design, or redesign, or the management of an MS. As already specified in Section 1.4, we adopt three modes for decision making (descriptive, normative, and prescriptive) depending on the system complexity and decision problem. However, regardless of the decision mode, the objective is to obtain a Decision Support System (DSS) capable of assisting decision makers. Section 1.3 already identifies the interest of Enterprise Modeling (EM) frameworks to implement such DSSs. One of the main advantages of these frameworks is the capability to consider in the same model different aspects of MS based on the *view* concept Subsection 1.3.1.

Since the ultimate goal of this work is to propose a methodology that allows to create a DSS that fits the Industry 4.0 environment, it is necessary to take into consideration two possible scenarios: the *greenfield scenario* and the *brownfield scenario* (see Subsection 1.2.2). In other words, either we create a new industry 4.0-oriented factory, or *smart factory*, or we operate a transition from an existing factory to a smart factory. In both case, taking into account the results of the old scenario is necessary. In this chapter, we review the decision problems in MSs and their evolution in the industry 4.0. First, in Section 2.2, we identify existing decision problems in MS in each activity.Understanding what existed before the new industrial revolution is a basis for studying the evolution of decision making. Our analysis of decision making in MSs

focuses on three main *plant activities*, namely, production, supply, and storage. In Section 2.3, based on the analysis of Chapter 1 we put forward the significant transformations of the old scenario (MS before the industry 4.0 revolution) into the smart factory. We identify three main evolutions, namely, technical, organizational, and transversal. For instance, with the advent of Industry 4.0, MSs are equipped with new technologies (e.g., IoT, M2M, AR), which transform them into smart MSs, thus, increasing the potential of the decision-making (Lu, 2017). Section 2.4 therefore highlights the evolutions of decision problems in MS within Industry 4.0, mainly concerning real-time decision-making, the change of the time horizon of some decisions, and the evolution of MS type.Finally, Section 2.5 concludes the chapter.

2.2 Decision Problems in Manufacturing Systems

As said, we focus on three MS activities: production, supply, and storage. We consider only the problems involved in the implementation of the plant, and assume that decisions related to the choice of products and processes have already been taken upstream. Therefore, decisions such as process planning, demand forecasting, and level of automation are disregarded. Furthermore, problems that do not belong to the three MS activities aforementioned will not be included in the scope of this work. Sourcing and distribution are examples of such activities.

In the following, we define decisions related the three MS activities.

2.2.1 Decisions related to Production Activity

This activity groups together decisions related to the design and management phases. The former phase contains the production strategy, the capacity planning, the line configuration, the line layout, and the line reconfiguration, while the latter includes the maintenance and process control, the lot sizing & scheduling, and the sequencing & scheduling.

2.2.1.1 Production Strategy

Production strategy is a strategic problem concerning the choice of the production mode. Based on the concept of the customer order decoupling point (CODP), Olhager (2010) distinguishes four strategies, namely Make-To-Stock (MTS), Make-To-Order (MTO), Assembly-To-Order (ATO), Engineering-To-Order (ETO) (Atan et al., 2017). In addition, this problem can also be considered in the supply activity, as under some production strategies, a balance between demand and supply can be required to avoid the left-over inventory or the shortage inventory.

2.2.1.2 Capacity Planning

As specified by Martínez-Costa et al. (2014), capacity planning (CP) problems encompass decisions related to the strategic, tactical, and operational levels. All these problems are considered belonging to the design phase.

Capacity investment: at the strategic level, the capacity investment decision problem predicts future production mix and volume based on marketing parameters. It minimizes the capacity invested, or maximizes the utilization subject to finite budget, capacity resource, and inventory balance. Van Mieghem (2003) conducts an extensive survey on optimal capacity investment problem.

Aggregate planning: at the tactical level, capacity planning is named aggregate planning. It addresses material requirements planning by determining the production level, staffing, and inventory on part families. Chakrabortty and Hasin (2013) propose a multi-period model for the aggregate planning by considering a forecasted demand and capacity for multiple products.

Capacity allocation: at the operational level, capacity planning deals with matching resource availability and capacity requirements for a production plan. The term capacity allocation is also often used to refer to this decision problem (Toktay and Uzsoy, 1998).

2.2.1.3 Line Configuration

Line configuration encompasses two strategic problems. On the one hand, there is the choice of type of assembly line to specify the model type (Single model, multi-model or mixed model), the assembly strategy (synchronized or postponement assembly), and the type of line automation (such as manual and robotic assembly). On the other hand, there are the assembly line balancing problem (ALBP), the equipment selection problem (ESP), and the assembly line design problem (ALDP), consisting of tackling the ALBP and the ESP jointly.

Choice of type of assembly line: Thomopoulos (2014) discuss the choice of type of assembly line. The authors describe in detail the type of assembly systems but do not specify how to choose between them. Based on their study, it can be stated that the choice of assembly line depends on many parameters such as production volume, variety of products, process planning, and level of automation. Ullah et al. (2014) review assembly lines and propose a classification based on different characteristics.

Assembly line balancing: the ALBP consists of assigning a set of production tasks to different workstations in order to optimize one or several objectives (Becker and Scholl, 2006). The study of Battaïa and Dolgui (2013) proposes a taxonomy of studied ALBPs. The work of Boysen et al. (2007) proposes to identify ALBPs by means of a tuple (α, β, γ) referring to (precedence graph characteristics, station and line characteristics, objectives) respectively. The graph precedence characteristics refer to the structure of the precedence graph and the processing times of each task. The station and line

characteristics refer to the physical layout of the assembly line, resource assignment, and other aspects of line configuration. Finally, the optimization objective evaluates solutions and can include single or multiple objectives, such as minimizing cycle time, idle time, or maximizing efficiency. Finally, Boysen et al. (2008) present an investigation about mathematical models used to deal with ALBPs in the real world and provide a structure of studied assembly systems.

Equipment selection: the ESP relaxes the assumption of identical workstations since each workstation could be equipped with its own equipment or share some with other workstations. Wilhelm (1999) considers the ALBP simultaneously with the ESP called the ALDP, taking into account the sequence of operations at each workstation. Özdemir and Ayağ (2010) propose an Integer Linear Programming to deal with a multi-objective case of ALDP.

2.2.1.4 Layout Design

In the production activity, we divide the layout design into Facility Layout Problem (FLP) and Line Layout Problem (LLP). Analogously, De Koster et al. (2007) study the layout design in the storage activity and subdivide this problem family into two problems. The facility's layout contains the order-picking system (OPS) and the layout within the OPS (internal layout problem).

Facility layout: in the design phase of MSs, the FLP is a critical strategic problem. It seeks to find an optimal disposition of different facilities within a plant (Saraswat et al., 2015). In the literature, depending on the material flow between facilities, Kulturel-Konak (2007) distinguishes static and dynamic versions of FLP. FLP can impact other decision problems: for example, to choose the MS type, FLP imposes constraints that delimit the space allocated to each zone in the plant.

Line layout: this problem is considered at the tactical level to decide the disposition of machines, taking into account the constraints imposed by the facility layout and the material flow between machines. Authors are often interested in minimizing the workflow interference or minimizing the total design cost. This problem is strongly linked with ALBP as we will see later. Ko et al. (2005) and Battini et al. (2007) review the most commonly used line type and line features in literature; among the most common line type we find such as straight line (SL), and SL with parallel stations that is often integrated in equipment selection problem.

2.2.1.5 Line Reconfiguration

Line reconfiguration is considered at the tactical level. According to the literature, the principal decision of the line reconfiguration is the rebalancing problem. However, some works point out that the rebalancing decision involves considering a new layout, giving rise to what is called the relayout problem.

Rebalancing: the rebalancing problem occurs when there are changes in the production process or demand, which requires the reorganization of the workstations or reviewing the assignment of tasks to workstations subject to the current line balancing constraints. To deal with the rebalancing problem, Boysen et al. (2007) propose two approaches. The first considers on the model the restricted assignment, while the second exploits the cost model as a criterion of rebalancing. Koren et al. (2016) consider the rebalancing problem in RMS, and they propose a mathematical model in the case of demand changes in order to minimize the number of machines while keeping the system throughput at the required level. Recently, Cerqueus and Delorme (2023) have shown that the rebalancing in RMS is not the most efficient policy when only the production volume change.
Relayout: as mentioned before, deciding the rebalancing may imply rethinking from scratch the line layout. Kovács and Kot (2017) refer to this problem as a relayout design problem and propose a new arrangement of machines within a plant by considering the current line layout as an input. The authors also imagine different possible scenarios for redesigning the layout taking into account the space constraint of the shop floor.

2.2.1.6 Lot Sizing

The lot sizing problem is widely studied in the literature concerning production planning (Martínez et al., 2018). This problem is not exlusive to the production activity but can also be met in the supply activity. It involves determining the optimal quantity of items to be produced in a production run over a specified time period, given certain constraints and objectives. Ramezanian et al. (2013) state that the goal is to determine the optimal production amount as a tradeoff between production, inventory, backorder, and setup costs.

2.2.1.7 Sequencing and Scheduling

On the one hand, the sequencing decision problem in production activity involves determining the order in which tasks or jobs must be processed in an MS to achieve a specific objective. On the other hand, the scheduling decision problem in production activity involves determining the start and finish times of each job or task to optimize a specific objective. Typical objectives include minimizing the total production time, maximizing the throughput, minimizing the number of late jobs, or minimizing the cost of production. The two decision problems are encountered in various production routing strategies, including Flow-Shop, Job-Shop, and Open-Shop. Therefore, the approach to solving them will depend on the specific shop type.

The scheduling problem involves the sequencing decision implicitly, and using the term scheduling will include not only the timetable decisions but also the sequencing decisions, i.e., order of launching products, models, end-items, or jobs. For the flow shop, all products are supposed to be manufactured in the same way on the same machines.

However, for the job-shop and open-shop, the routing of each product could be individual. Furthermore, the amount of setups is not negligible since the setup of machines takes place very often.

When it comes to solving the sequencing and scheduling problems, the MS type must be considered. The dedicated manufacturing system (DMSs) is designed for a single product and follows a flowshop production routing strategy. Therefore, sequencing the products is irrelevant in DMS, and the scheduling problem is addressed without considering product sequencing, as noted by (Gowrishankar et al., 2001). However, Flexible Manufacturing Systems (FMSs) or Reconfigurable Manufacturing Systems (RMSs) allow the production of multiple models in the same line or shop, either in batches (multi-model) or one-piece (mixed-model) production. Hence, in this case, the product sequencing problem becomes critical to consider, as it can impact productivity through machine idle time and costs due to penalties for missed deadlines.

2.2.1.8 Maintenance and Process Control

Maintenance is functional to the production process, as its objective is to keep the production system in a state that allows it to perform its function (Budai et al., 2008). Process control refers to the process inspection intervals and consists in performing the maintenance schedule to keep as long as possible the system in a state in which is under control. The maintenance and process control include five decision problems whose time horizon can differ.

Maintenance strategy selection: at the strategic level, this problem consists in deciding which strategy or policy of maintenance to use for each equipment of an MS, taking into account budget constraint (Bevilacqua and Braglia, 2000). Several strategies exist that are derived from the two most known ones, i.e., corrective maintenance and preventive maintenance. Recently, Meng et al. (2021) tackle ALBP taking into account preventive maintenance scenarios, and point out that in the case of corrective maintenance strategy if a workstation fails suddenly, rebalancing is the only way to recover production.

Maintenance load forecasting: at a more tactical level, we address the problem of maintenance load forecasting, which consists in predicting the maintenance load taking into account the preventive replacement threshold (Duffuaa and Raouf, 2015). The maintenance strategy, the replacement costs, and the average deterioration speed are among the factors that can impact the forecast.

Maintenance capacity planning: as specified by Duffuaa and Raouf (2015), maintenance capacity planning occurs at the tactical level once the maintenance load is forecast. This problem consists in determining the skills and the number of resources needed to meet the demand for maintenance work while maximizing the resource utilization or minimizing the total cost.

Maintenance scheduling: it is an operational problem that specifies how, when and where

the maintenance duty is performed. Given the maintenance tasks to be accomplished at given dates, the objective is to provide the maintenance schedule, over a given planning period. By assuming that the maintenance is always done at the end of a production cycle, Lee and Rosenblatt (1987) solve the maintenance scheduling and lot sizing problem (see Subsection 2.2.1.6) jointly. The authors conclude that the optimal lot size and maintenance schedule is a tradeoff between setup, holding, and inspection costs.
Maintenance control: according to Gits (1994), maintenance control is concerned with the operational coordination of demand and resources in order to reduce as much as possible the unavailability of the plant. The maintenance management system is often required to address this problem. The aim is to revise the maintenance strategy, upgrade work orders, change the maintenance schedule, and propose training of resources.

2.2.2 Decisions related to the Supply Activity

The supply activity concerns the configuration feeding system, the choice of the line feeding policy, the inventory, and the loading of components from storage zones to the lines. The efficiency of the MSs is affected by the structure of parts feeding systems that are composed of three main components: storage of parts, transport of parts, and feeding policy (Kilic and Durmusoglu, 2015). In the last decade, many papers have studied the problems related to line feeding.

The work of Emde and Boysen (2012b) classifies decision problems related to the planning and control of line feeding into five decision problems: location planning of feeding system, sizing problem, handling operators routing problems, scheduling problem, and loading problem. Recently, Schmid and Limère (2019) propose a classification of different line feeding problems based on new notation (α, β, γ), in which α is used for the descriptions of line feeding policies, assembly line configurations, and product characteristics. The β encompasses the assumptions and decisions that govern the entire assembly line feeding process, such as the number and locations of supermarkets, material flow, and automation level of the preparation process. Finally, the γ refers to the objective functions.

2.2.2.1 Feeding System Configuration

In the feeding system configuration, we include four main problems, two strategic decisions (location planning of feeding system and sizing) and two tactical decisions (routing and scheduling). On the one hand, Faccio et al. (2013) state that location planning of feeding system, sizing, and routing problems are taken at the strategic level. On the other hand, many works (Alnahhal and Noche, 2015a; Emde and Boysen, 2012b; Emde and Gendreau, 2017) consider the sizing and routing problems jointly. In our study, we choose to consider each one independently because we think that the sizing problem is a strategic decision, but the routing problem could be considered either at

the tactical or operational level.

Location planning of feeding system (LPFS): the aim is to determine the location and size of *supermarkets* and specify the parts each one should handle. Supermarkets are specialized storage areas within an MS where parts are stored in sufficient quantities to support the production process. They are typically located close to the assembly line. Battini et al. (2010) deal with the LPFS using the concept of supermarket and consider that only one supermarket could serve the assembly line, i.e., feed the line with the required parts when needed, while Emde and Boysen (2012a) solve it when several supermarkets could serve the components to the same assembly line without restriction about their locations around the stations.
Sizing: the sizing or fleet size problem seeks to find the optimal number of milk runs or vehicles to feed the MS. By considering the stations to be supplied with parts, Emde and Boysen (2012b) try to minimize the number of vehicles and stock at the border of lines ensuring no line stoppage.
Routing: this problem seeks to find the trip of each operator, i.e., the starting and ending point for each vehicle. Satoglu and Sahin (2013) develop a mathematical model and a heuristic to solve this problem. The authors consider different feeding policies and construct the routes to deal with a just-in-time periodic feeding.
Scheduling: Golz et al. (2012) state that the scheduling task of the part feeding problem is similar to the vehicle scheduling problems. It consists of assigning tasks to vehicles while complying to the starting and ending times of the tasks and travel times from one task to another. The goal is to minimize the labor costs using this assignment.

2.2.2.2 Line Feeding Policy

In the literature, four policies are mainly considered for the feeding of the line, namely line stocking, Kanban (Continuous supply), sequencing, and kitting (Battini et al., 2009; Sali and Sahin, 2016). Battini et al. (2009) propose a framework to design the optimal feeding policy w.r.t. the total time spent to feed the assembly line. They state that several factors impact the efficiency of assembly system design and decision on the right feeding policy. Among these factors are the distance between the warehouse and the assembly line, the average number of components for each final product, the average assembly lot size, and the number of workstations.

2.2.2.3 Inventory

The inventory problem does not belong only to the supply activity, but also to the storage activity. We review here the two main inventory-related problems, i.e., the inventory policy problem and the inventory planning and control problem. The inventory policy problem is considered at the tactical level, while the inventory planning and control

problem is considered at the operational level.

Inventory policy: two different inventory policies are commonly used in the literature for replenishment: the continuous review policy in which the inventory is revised every day and the decision of whether and how much to order is taken accordingly; and the periodic review policy, in which the inventory level is checked at regular intervals, and an appropriate quantity is ordered after each review (Singha et al., 2017). Recently, Toha et al. (2019) present a comparison of the two policies based on a case study. They express the total cost generated by each policy and demonstrate that using the continuous review outperform the periodic review in their case.

Inventory planning and control: this problem involves determining the optimal quantity and timing of inventory to be stocked by a company to meet demand, while minimizing costs associated with inventory holding, ordering, and shortage. Commonly, depending on inventory policies, two ordering approaches are used for the inventory planning and control. The (R, Q) approach consists of ordering the quantity Q each period review R, while the (s, S) approach consists of ordering the quantity S when the stock level achieves the threshold s (Perera et al., 2017). Recently, Nahmias and Olsen (2020) discussed in detail different models to deal with inventory control (tracking and adjusting the inventory level) either under deterministic or stochastic demand.

2.2.2.4 Loading

The problem of loading is considered at the operational level, and consists of determining the number and type of components to be loaded per trip, i.e., tour of a tow train (Emde et al., 2012; Golz et al., 2012). The goal is to minimize inventory at the border of lines in each station, ensuring that the line will not be stopped due to components shortage. Faccio et al. (2013) use the supermarket concept and propose an integrated approach. They solve the sizing problem and exploit simulation to deal with the loading problem based on an analytic model. They consider Kanban as a line feeding policy and take into account the capacity of each tow train, the refilling intervals, and the number of tow trains.

2.2.3 Decisions related to the Storage Activity

In the production warehouse used in an MS plant to store the components, semi-finished and finished products, three types of areas are commonly found: the reserve storage, the forward and cross-docking areas (Heragu et al., 2005). A typical example of a reserve storage area in a plant is where components are stored after reception from a supplier or stored after production. Supermarkets are an example of forward area. The classical form of the warehouse has parallel aisles with products stored alongside. Generally, we quantify these products in stock-keeping units (SKUs), i.e., the smallest unit of load manipulated in the warehouse to manage the orders.

Generally, the storage activity considers the warehouse design and warehouse management (operation) problems (Gu et al., 2007). The review of Gu et al. (2010) covers the warehouse design as to the overall structure, the sizing and dimensioning, the equipment selection, the department layout, and the operation strategy selection. Following their work, we only consider three of the five design problems at the strategic level. Therefore, we do not consider the overall structure from our study and consider the operation strategy selection as part of the warehouse management at the tactical level. This choice is consistent with that of other authors in the literature, such as Rouwenhorst et al. (2000) or Gagliardi et al. (2012).

For the warehouse management, we conside: at the tactical level, the operation strategy, zoning, and Storage Location Assignment Problem (SLAP); at the operational level, the order picking, which groups three operational decisions: batching, routing and sequencing, and workforce assignment.

2.2.3.1 Warehouse Dimensioning and Sizing

Gu et al. (2010) define the sizing problem as the determination of the storage capacity needed at each period. The dimensioning problem consists of finding the optimal storage space (to built and/or to lease) to satisfy the required storage capacity. Most of the time, the decision of leasing a space (contract warehousing) at each period is considered when dealing with the dimensioning problem (Cormier and Gunn, 1996; White and Francis, 1971).

2.2.3.2 Equipment Selection

The equipment selection in storage activity differs from this of production activity due to mainly the objectives, the main features of equipment, and the decisions factors. In production, the equipment needs to be capable of performing specific production tasks, such as assembling, or processing raw materials. It may involve machines, tools, robots, or specialized equipment depending on the production requirements. However, in storage, the equipment needs to be capable of storing and retrieving goods. It may involve forklifts, conveyors, or automated storage and retrieval systems (AS/RS).

We have considered two major strategic problems in the equipment selection problem of the storage activity: storage equipment selection and material handling selection. We did not include the level automation decision in this problem family as Gu et al. (2010) did in their work because this decision is out of our thesis scope and usually uses the designer's experience. Furthermore, few works exist in the literature addressing this problem.
Storage equipment selection: Groover (2016) classifies the storage equipment into two categories, the traditional and automated storage equipment. Many traditional storage methods and equipment make use of manual picking solutions. They are chosen in the warehouse depending on the products. Automated Storage/Retrieval Systems (AS/RS)

are among the most popular automated storage equipments. It is a product-to-picker system used on one or multiple aisles. Each aisle is usually equipped with a robotic crane or storage/retrieval machine to retrieve or store the product on the racks of the aisle. For more details about AS/RS, we refer the reader to Roodbergen and Vis (2009).
Material handling selection: according to Mohsen (2010), Material Handling Equipment (MHE) selection is one of the most critical decisions in the design of the Material Handling System (MHS). MHE consists of determining the type and number of equipment as well as their specifications. Chakraborty and Banik (2005) consider four main MHE groups and uses the Analytic Hierarchy Process (AHP) technique, which breaks down the problem into a hierarchical structure of criteria and sub-criteria and assigning relative weights to each. Recently, Bouh and Riopel (2016) present a new classification of MHEs and identify seven categories: each one is subdivided into classes first, then into types.

2.2.3.3 Department Layout

The department layout is concerned with decisions about the shape and configuration of the warehouse and includes three main strategic decision problems: storage layout, block-stacking, and configuration of AS/RS.
Storage layout: the Storage Layout Problem (SLP) consists in determining the warehouse shape and the location of its elements. It includes several decisions, such as optimizing the number of aisles, defining their length, width, and orientation, determining the door and Pick and Deposit Point locations. SLP is usually the decision problem which is tackled first in the department layout problem family. Cardona et al. (2015) cope with the new layout called the *fishbone layout* introduced by Gue and Meller (2009). The authors propose a method to generate the warehouse layout with unit-load single command operations.
Block stacking: this problem aims to specify the number of lanes and their depth, the stack height, and decide the pallet placement angle regarding the aisle with the storage clearance between pallets. The aisle orientation determined by the SLP is considered an input (parameter), and the warehouse height is a constraint. Derhami et al. (2016) propose new mathematical models in form of algebraic expressions to compute the optimal lane depth with the aim to maximize utilization of the volume.
Configuration of AS/RS: this configuration is determined at the strategic level of warehouse design if AS/RS is selected as the storage equipment. Roodbergen and Vis (2009) consider two main aspects of the physical design of AS/RS, the system choice and the system configuration. As to system choice, authors provide inputs and constraints, then summarize outputs, i.e., decisions of this problem. In our study, the number of aisles, their length and width decisions are supposed related to the SLP, as many authors stated (Gu et al., 2010; Roodbergen et al., 2014). Besides, this decision problem is made before the configuration decision. Based on the handling operation of

picking process, Groover (2016) distinguish seven types of AS/RS.

2.2.3.4 Operation Strategy

Under the name operation strategy, go all the decision problems that define the rules for carrying out the three main processes of this activity, i.e., storage, routing, and picking.
Storage strategy: the storage strategy (policy), sometimes called the storage assignment policy, is a set of rules to follow to find the location of each SKU in the warehouse. Ang et al. (2012) group all the storage strategies discussed in the literature into two categories: dedicated storage strategies and shared storage strategies. The former category assigns a fixed location to each SKU type and typically uses the Cube Order Index (COI) rule. Shared storage strategies allow a more flexible use of space. Among them, the random storage strategy that assigns SKUs to empty locations randomly, as it considers no assignment restrictions, and assumes that all locations could store all SKUs.
Routing strategy: similarly to the storage strategy, the routing strategy is a set of rules to follow by an order picker to organize the tours from an input/output point to locations to retrieve the products. The main objective is to minimize total material handling costs and expected travelled distance. The recent review of Masae et al. (2020) shows that 77% of the papers use heuristic routing strategies. The remainder instead exploits exact and metaheuristic methods. Several routing strategies have been studied in the literature, namely traversal, midpoint, largest gap, return, composite, and S-shape (Petersen, 1997).
Picking strategy: the picking strategy is also a set of rules to organize the order picking process. Five main picking strategies (single order picking, batch picking, zone picking, wave picking, bucket brigade picking) have been studied in the literature with multiple variants (Parikh and Meller, 2008). Petersen II (2009) evaluate different picking strategies using a simulation approach. They assume a specific storage strategy and storage layout with a fixed zone configuration.

2.2.3.5 Storage Location Assignment Problem (SLAP)

In this problem, the SKUs of orders are assigned to locations (e.g., zones, slots) in the warehouse. Static and dynamic versions of this problem can be found in the literature. In the former version, the assignment does not change in each period, and rolling horizon over the periods to assign the items is not considered, and according to Gu et al. (2007), it is the version often studied in the literature. This static version of SLAP could be considered at the tactical level. In comparison, the latter version considers a planning horizon from period to period to decide the assignment of new customer orders and which items to relocate before the start of a new period. This version is more complicated and depends on several characteristics such as demand forecasts, the distance between the current and future storage locations of items, zone structure. According to Rojas-Reyes et al. (2019), this problem is often tackled at the operational level.

2.2.3.6 Zoning

The zoning problem divides the picking area into picking zones and organizes the order picking process by zones. We consider here the zoning size problem at a tactical level (De Koster et al., 2012), whereas the zoning assignment problem (workforce allocation) is made at an operational level according to (Van Gils et al., 2018).

Zoning size: this problem concerns the determination of the number of zones and pickers (called *workforce level* by Van Gils et al. (2018)), and the storage space per zone. De Koster et al. (2012) tackle the problem of determining the optimal number of zones in order to minimize the throughput time. They assume the batch picking strategy under parallel zoning with sort and pack-after-pick. In the literature, few articles tackle the problem of the optimal number of pickers to use in each zone. Recently, Van Gils et al. (2016) use a time series to predict the number of daily pickers required to fulfill the orders.

Zoning assignment: the zoning assignment problem concerns the assignment of pickers to zones. Obviously, the number of zones and the number of pickers are considered as inputs of this decision. Research on this problem is still scarce. Based on the queuing models, in the case of progressive zoning (the orders are picked zone by zone, i.e., the order picker picks all orders related to its zone, then pass the container to the next zone and so on (pick-and-pass)), Van Nieuwenhuyse and De Koster (2009) study the impact of workforce allocation to picking and sorting operations on the average customer order throughput time. On the other hand, De Koster et al. (2012) investigate the case of parallel zoning (the picking in each zone is done in parallel) for the pick-and-sort system. However, they consider the total number of order pickers fixed and spread the same number of pickers over zones.

2.2.3.7 Order Picking

Order picking groups together decisions about the order picking process. We describe three main decisions, namely order batching, the routing and sequencing, and the workforce assignment.

Order batching: the Order batching Problem (OBP) specifies the orders for picking in a single tour by combining them into one picking list. In the literature, we distinguish two types of batching, the static and the dynamic batching (Yu and De Koster, 2009). The former assigns each order to a batch given the order lines at the beginning of the planning horizon. The latter consists of deciding the batch size or the time interval to batch orders (batch time window) to minimize the average throughput time. Henn et al. (2012) provide an investigation of batching methods and highlights the fact that most of the known approaches are heuristic. A recent literature review on order batching problem is given by Cergibozan and Tasan (2016).

Routing and sequencing: this problem concerns the essential decision of the order picking

system at the operational level. It consists of finding the sequence and route for picking or storing a set of SKUs based on their current locations or predefined ones. De Koster et al. (2007) state that commonly the main objective is to minimize the travel time or travel distance. When considering batching, the sequencing decision is to find the sequence and derive the possible batch completion time to minimize the total tardiness of all orders. On the other hand, the routing decision takes care of the sequencing of each batch and, based on the SKUs assigned to it, determine the batch completion time and associated travel time. Sometimes in the literature, the routing and sequencing problem is referred to as the Order Picking Problem (OPP) (Çelk and Süral, 2013).
Workforce assignment: we consider here the Workforce Assignment Problem (WAP) at the operational level. This problem consists in assigning the orders to the order pickers. Most of the time, the dynamic version (called the online picking problem by Rubrico et al. (2011)) is considered rather than the static version, which requires all orders to be known beforehand. Usually, the goal is to minimize the picking time of the entire schedule. Links exist between the WAP and previous problems belonging to the warehouse activity (picking strategy, zoning assignment, batching, routing, and sequencing). In addition, the WAP entails constraints such as picker capacity, picker skills to avoid picker blocking and thus reduce labor costs (Hong et al., 2012).

2.3 Transformation of the Old Industrial Scenario into Industry 4.0 Scenario

Today, the rationality of decisions in MSS has changed due to significant evolutions. An analogy with a metaphor on the choice of transport mode can be helpful. In the past, buying a car was often the most convenient solution for one's daily trips to the workplace. Nowadays, many more transport modes are available, and many service-oriented possibilities exist, like leasing, car-sharing, or taxi booking services, proposing real-time prices according to the customer location and the nearest taxi. Therefore, depending on factors such as covered distance, or frequency of trips. One can choose, each time, the customized travel solution that better fits the specific circumstances. In the same way, in the past, MSs were much less flexible due, for example, to technological barriers that prevented real-time decision-making. However, today in smart factories, more options are available to companies, directly impacting the priority and the interactions of the decisions. For instance, by leveraging advanced automation technologies, companies can optimize their operations and enhance their decision-making capabilities. The maintenance control could be done in an optimized way in accordance with the production planning decision based on real-time monitoring. Additionally, companies can benefit from the new data analytics tools to extract insights and make data-driven decisions.

We resort to the literature to determine how Industry 4.0 meets the challenges identified in Subsection 1.2.2, namely: short product life cycle, complexity of the process, integration of data, implementation of new technologies, evolution of business models, integration and development of new skills, and development of sustainable MS. From the decision-making perspective, rising to these challenges means defining a new decision paradigm because new constraints must be considered (e.g., integrating two or more decisions or considering new decision interactions). Also, the temporality criteria must be considered in studying and solving decisions (e.g., real-time decisions), and the emergence of new types of MSs must be investigated to determine how to support decision-making.

In the literature, some tentatives start to emerge to face these challenges, however, without standardization at present. In the following, we will present a short literature review that shows an analysis of research works conducted for this purpose.

On the one hand, decisions now rely more on data-driven approaches, which imposed the integration of data challenge. A standardization of data exchange is one of the potential answer to ensure the connectivity and integration and making data available throughout the entire value chain of the enterprise. However, this integration of data gives rise to a huge amount of data either in the greenfield or brownfield scenario, which means data structure may be standardized or not. Furthermore, the necessity of data processing even in near real-time may complexify the decision-making. One of the potential answer to face this complexity of decision-making induced by the huge amount of data is to develop new techniques and algorithms, mainly to enable the resolution of decision problems requiring high computational resources.

On the other hand, new market needs such as personalization leads to the challenge of developing new business models. A potential answer is to adapt a leagile strategy, i.e, focus on add value processes and be agile. Within Industry 4.0 collaborators, such as workers, are no longer mere executors but decision-makers and problem-solvers in their own right, which imposed a critical challenge of integration and development of new skills. For this reason, a potential answer to meet this challenge is to develop a human-centered MS. Decision making faces the challenge of uncertainty in the Industry 4.0 context. RMSs were traditionally developed to address this challenge with their six main features. However, RMSs are designed with a centralized control approach and do not incorporate sustainability considerations. Some recent efforts explore a decentralized control approach and sustainability dimension in RMSs, but a consensus has yet to be reached. One potential solution to tackle uncertainty is to upgrade RMSs and integrate these aspects into a new MS type called Adaptable MS.

Finally, an innovation-driven production and high competitiveness imposed the short product life cycle challenge, which means also a product maturity must be achieved rapidly. One of the potential answer to this challenge is a real-time monitoring during the design, production and usage of product to accelerate the learning time of product

behavior and its performance. With the short reconfigurability time required, the complexity of process increased and one of the potential answer proposed to handle this complexity is to use a decentralization control approach in MSs. Also, new regulations and market trends imposed the development of sustainable MSs. It means the way of MSs assessement is changed, leading to a revision of the criteria for decision-making. That is why, a potential answer of this challenge is the use of sustainability-oriented KPIs for the MS assessement.

All these potential answers presented are considered as precursors for leading three significant evolutions, namely: technical, organizational, and transversal, as shown in Table 2.1 and will be explained in Subsections 2.3.1 to 2.3.3. The classification of these evolution originates from our interpretation of the work of Erol et al. (2016). The authors state that the vision of Industry 4.0 leads to technical and organizational complexities that impose many challenges. We add to technical and organizational evolutions what we call transversal evolution, i.e., those which involve both technical and organizational aspects. For instance, using agility as a sustainability-oriented KPI gives rise to model changes and even algorithms. This is because new objectives must be considered, which can result in multi-objective problems or the introduction of non-linearity (e.g., taking energy consumption into account when scheduling production). At the same time, agility requires adapting new business models leading for example customers to require specific performance of product carbon footprint, then impacting the pricing. Therefore, considering a sustainability-oriented KPI will involve both technical and organizational aspects.

Definition 2.1 *A transformation driver is defined as a potential answer to address a challenge of Industry 4.0, and as a consequence, it serves as a precursor for leading either the technical, organizational, or transversal evolution from the old scenario to the Industry 4.0.*

In the following, in this manuscript, when we mention a *transformation driver*, we will refer to this specific definition. In Table 2.1, there are three columns describing the challenges, their corresponding main transformation drivers, i.e., the potential answers to meet the challenges, and the type of the corresponding evolution.

Table 2.1: The challenges that Industry 4.0 must face from a decision-making perspective.

Challenges	**Main transformation drivers**	**Evolution type**
Integration of data	Standardization of data exchange New techniques and algorithms	Technical
Evolution of business models	Leagile strategy	Organizational
Integration and development of new skills	Human-centered MS	
Uncertainty	Adaptable MS	
Short product life cycle	Real-time monitoring	Transversal
Complexity of the process	Decentralized control approach	
Development of sustainable MS	Using KPI 4.0	

From now on, we characterize the new decision paradigm through technical, organizational, and transversal evolutions.

2.3.1 Technical Evolution

In Industry 4.0, decisions become data-based and require standardization to avoid interoperability issues. As an example, a decision made at a specific moment may become inefficient an hour later. Moreover, new adapted algorithms are required to deal with such massive data generated by the implementation of new technologies.

In what follows, we will detail the transformation drivers of the technical evolution, namely, standardization of data exchange, and new techniques and algorithms.

2.3.1.1 Standardization of Data Exchange

In Industry 4.0, smart devices make data available anytime, anywhere. However, only data that can be transformd in meaningful information is actually useful for decision-makers. When an MS enables data-driven decisions, integrating data in this MS can be considered successful.

However, in Industry 4.0, in order to achieve data-driven decisions, data must go from the connection level to the cognition level of 5C architecture (see Subsection 1.2.3.3). This requires a standardization to organize the exchanges between systems based on reference data models and suitable data structures, otherwise interoperability issues can emerge and prevent data integration correctly.

The classical approach aiming to cover data integration in the old scenario was based on the automation pyramid promoted by ISA 95, which consists of five levels:

1. the field level involves sensors, devices, and actuators on the production floor that perform physical work and monitoring.

2. the control level uses programmable logic controllers to control the field devices.

3. the supervisory level uses supervisory control and data acquisition (SCADA) systems to monitor and control multiple systems from a single location.

4. the planning level uses a computer management system known as Manufacturing Execution System (MES) to monitor the entire MS process in a plant, from raw materials to the finished product.

5. the management level uses an enterprise resource planning (ERP) system that integrates all levels of operations of a company, from manufacturing to sales, purchasing, finance, and so on.

The hierarchical structure proposed by ISA 95 becomes rigid with the introduction of the decentralized control approach because any change in one level necessarily impacts the other levels. Therefore, the data flow must be readapted.

Efforts are made to enhance this drawback. For instance, IEC (2015) is among the standards proposed to specify the functional requirements for manufacturing operations management (MOM) systems. This standard defines the common data model and data exchange formats for MOM systems to ensure interoperability between different systems. Additionally, it outlines the requirements for data security and integrity, as well as the necessary procedures for system validation and testing.

A 5C architecture proposed in the Industry 4.0 scenario for CPSs also struggle with the hierarchical structure issue. Therefore, the OPC UA Foundation proposed a standard of data exchange based on a model of client and server called OPC UA client and OPC UA server, respectively.

The OPC UA server can describe the data of each physical asset in the plant by using flexible data modeling, in which each object or data is represented as a node and a tree structure is used to link these nodes. Likewise, the server adopts a service-oriented architecture (SOA) to expose its services to the OPC UA client, e.g., the read data service and write data service to respectively access and modify specific object tags. Thanks to these services, the client can consume the data from the server. Moreover, both client and server can be deployed across different platforms.

To have more flexibility in terms of data management, some recent solutions, such as MQTT or SparkplugB, use a publish/subscribe pattern rather than the client/server pattern. In it, nodes interested in specific tags subscribe to the corresponding topics and are notified each time tag values change.

Thanks to solutions as e.g., OPC UA, MQTT, SparkplugB, the challenge of integrating data can be met and the readability and reusability of data improved.

2.3.1.2 New Techniques and Algorithms

As already stated in Subsection 1.2.2, greenfield and brownfield are possible scenarios to implement new technologies.

Many works in the literature investigate the case of a brownfield scenario that consists of retrofitting existing equipment. For example, Lins et al. (2018) describe the retrofitting process in three parts: infrastructure, communication, and application. The infrastructure part concerns introducing new Industry 4.0-oriented equipment (e.g., IoT, cobot). The communication part copes with the integration of the protocols of industrial networks, especially those adapted for IoT networks, most often an OPC UA gateway since almost all manufacturers have an OPC UA interface as an integral part of their products (Etz et al., 2020). Finally, the application part supports the various industrial applications by using data cloud and web services.

In the case of a greenfield scenario, MS is built from the ground up, therefore, more options exist as to technical solutions w.r.t. the brownfield scenario, although this means higher investment costs. To compare the two scenarios, the reader could refer to Albo and Falkman (2020).

From the point of view of decision making, the effects of integrating data are mainly related to the communication and application parts than to the infrastructure part. An example is analyzing mass and ubiquitous data communicated by new or upgraded installations to use them for decision making. As a consequence, facing the challenge of integrating data leads also to the need of developing new techniques and algorithms capable of handling these new effects.

In what follows, we give the reader an overview of new techniques, algorithms, their changes and trends following the implementation of new technologies.

In practice, it is common to adopt multistage optimization to deal with variability, fuzzy set theory to cope with ambiguity, and robust optimization to deal with imprecision. In the same context, according to Csalódi et al. (2021), modeling the uncertainty cannot be avoided in Industry 4.0. The authors advocate using stochastic programming, fuzzy programming, or stochastic dynamic programming to deal with it. Likewise, they advise exploiting a modular structure of MS to reduce the complexity of models, then using multistage optimization techniques to solve the models of modules even in parallel.

Currently, exploiting the real-time data of MSs enables taking into account dynamics aspects in decision problems. For this purpose, two approaches that are becoming more and more popular are: the coupling of optimization and simulation models, and the hybridization of optimization and Machine Learning (ML) techniques.

New technologies contribute to handling uncertainty and dynamics in Industry 4.0 by allowing decision makers to monitor the MS state with high periodicity and make decisions using advanced AI techniques. Usually, these AI techniques are classified according to

two approaches: predictive and proactive. Predictive approaches can predict events in the future by training a model with a predefined dataset. In contrast, proactive approaches have more autonomy since they use prescriptive models capable of making intelligent decisions on their own, so as to suggest actions and outcomes.

ML, which falls under the broader umbrella of AI, has significant implications for MSs in Industry 4.0. Among the most known subcategories of ML, and more precisely of proactive approaches, are deep learning and reinforcement learning. Deep learning algorithms base their training on neural networks (Bishop, 1994). Therefore, they do not need structured data and guidance for making a decision. Recurrent neural networks, de-noising autoencoder, and long and short-term memory networks are examples of such algorithms (Mosavi et al., 2020). Reinforcement learning, influenced by behavioral psychology, involves an agent interacting with its environment, learning optimal strategies through trial and error (Kaelbling et al., 1996). Actions, determined randomly or based on past performance (e.g., Q-Learning algorithm, for details see Watkins and Dayan (1992)), impact the environment state, and their success is measured by the reward received. Examples of an agent and its environment are a robot and a plant: actions can be decisions related to the robot's movements, and the states are possible locations of the robot. The reinforcement learning uses a Markov decision process for decision-making (Nian et al., 2020).

In a related context, Spezzano and Vinci (2015) and Rai et al. (2021) acknowledge the role of ML algorithms, such as cluster analysis, for extracting valuable insights from data generated by physical devices. ML techniques are seen as crucial for detecting complex production patterns in smart manufacturing, providing timely decision support, and emphasizing the role of AI in real-time decision-making, even for Big Data. Finally, they present the four main applications of ML in manufacturing: computer vision-based inspection and monitoring, fault detection, cloud manufacturing, and process improvement and optimization. Examples of the techniques used in these applications are convolutions neural networks, support vector machines, decision trees, and process mining.

2.3.2 Organizational Evolution

New revenue models have emerged with the evolution of business models, such as dynamic pricing, pay-by-usage, or performance-based billing, i.e., the cost for product/service is based on its performance. In this regard, companies seek to adapt and optimize their MS using new organization modes, aware that they have to be flexible, i.e., capable of rapidly changing their mindset, practices, and configuration to meet customer requirements. This is what happens in *agile* MSs. At the same time, they need to be *lean*, and try to keep in their MS only processes that add value to the product/service the customer intends to pay. Combining lean and agility is called *leagility* (Mason-Jones et al., 2000), or *leagile*

strategy (Naylor et al., 1999). Implementing new technologies with the leagile strategy contributes significantly to the proposition of customized and innovative product/service solutions hence inscrease the offered value.

Another change at the organizational level is the development of new skills of employers due to the emergence of new technologies (Barbieri et al., 2020). As stated in Subsection 1.2.2, the integration and development of these new skills are one of the challenges of MSs in industry 4.0. To do so, upskilling or reskilling are preferrable to using only automation as this latter is not adapted for complex processes where broad flexibility is required. For instance, Bortolini et al. (2021) point out the increasing need for collaboration between people and automation technology in the context of Industry 4.0. They recommend an adaptive automation assembly system focused on the collaboration between humans and machines. Ejsmont (2021); Roux et al. (2017) consider that the human factor must be at the center of MS to succeed in the implementation of Industry 4.0 scenario. Moreover, creating a human-centered MS leads to a better working environment, improves the qualifications of employees, and contributes to the introduction of appropriate training programs.

Uncertainty in MSs can stem from various factors. Among others, these include the lack of visibility regarding the next product generation, unpredictable customer behavior patterns, uncertainty about production volume, the influence of evolving regulations on product sales. External factors such as geopolitical, socioeconomic, and pandemic crises can also significantly contribute to uncertainty. The level of adaptability required in MSs can vary, ranging from low to high. A low level of adaptability may involve adjusting production volumes in response to customer demand needs (El Alaoui El Abdellaoui et al., 2020). On the other hand, a high level of adaptability may necessitate a more substantial transformation, such as manufacturing repurposing that consists of redefining the core business and processes(Ho et al., 2022).

In the next sections, we present the leagile strategy, the concept of Human-centred MS, and the Adaptable MS as three organizational evolutions that allows to answer to the challenges posed by, respectively, the evolution of business models, the development of new skills, and the uncertainty.

2.3.2.1 Leagile Strategy

The concept of leagile strategy is a new way of implementing the postponement strategy, which is related to the concept of CODP (c.f. Subsection 2.3.3.1). It tries to face stable demand using the lean strategy for standard product, and demand volatility using the agility aspect for customized products.

The complementarity of leanness and agility concepts eliminates waste and allows a better reactivity of MSs. The work of Rossit et al. (2019) highlights that they are essentials for the MS organization in Industry 4.0 to face the evolution of business models. The

authors point out that MSs must offer the possibility to make decisions online to make possible for customers to make changes in their orders during the production process.

For instance, Bortolini et al. (2017) introduce a new framework to design assembly line systems with a new production strategy called personalized production. The objective is to propose a FaaS (Factory-as-a Service, see Subsection 1.2.2). They state that the availability of information of the production process in real-time allows customer integration from the product design stage to the product final assembly stage. This is made possible by cloud solutions that allow the customer to monitor the assembly process and modify the ordered product at each stage.

Companies are beginning to integrate the leagile strategy even in product development. For instance, Lemieux et al. (2015) provide a leagile transformation methodology as a guide for manufacturers for this purpose.

Furthermore, customer integration in the value chain enhances collaboration and enables a better customer experience. Based on this, an improved decision making becomes possible for what concerns product/service offers, business orientation, future market requirements, and production process. Therefore, we can state that leagility brings an excellent answer to the challenge of the evolution of business models. Including leagility into decision aspects from strategic to operational level allow maximum benefits for companies.

2.3.2.2 Human-Centered MS

In order to Build a human-centered MS, it is necessary to understand the job categories, skills, and management styles that better fit the Industry 4.0 scenario.

For this purpose, external organizations or research institutions nowadays develop and propose learning programs to help companies getting familiar with Industry 4.0 concepts (Maguire, 2015). The objective is to exhibit technologies, show challenges, and highlight competencies required to succeed in this transition. For instance, IT'm Factory at Mines Saint-Etienne is one of the examples of such a learning factory in France that serves as a training platform for students, researchers, and practitioners. Furthermore, according to the objective of each company, IT'm Factory experts propose a dedicated transformation roadmap with change management (Bennara et al., 2020; CFAI Loire and Institut Henri Fayol de Mines Saint-Etienne, 2018).

Volpentesta and Felicetti (2011) defines *competency* as knowledge and skills required to carry out a specific task, e.g., decision making efficiently. Based on it, Gudanowska et al. (2018) consider four categories of competencies in their work: social, personal, managerial, and professional. For example, decision-making is a part of personal competencies, which is considered necessary in Industry 4.0 by (Ejsmont, 2021). To perform the decision making task, DSS are crucial, as already discussed. Furthermore, according to Kaur et al. (2019), mastering DSS is among the computer skills that support Industry 4.0.

Based on this analysis, from a decision making perspective, human-centered MSs must aim to develop the personal competencies of each collaborator (e.g., worker, engineer, and managers), especially IT skills. This conclusion is coherent with the works of Fantini et al. (2020); Prifti et al. (2017), which state that workers become decision makers as well as problem solvers, and engineers have to collaborate with IT specialists and have an interdisciplinary competency such as modeling, optimization, and simulation.

Further, due to the importance of ICTs in the Industry 4.0 environment and its rapid evolution, decision makers are asked to change their learning attitude and switch to lifelong learning. This is becoming even more true since the management style changes from power-driven to value-driven (Erol et al., 2016). In addition, the classical top-down approach is more and more freqently replaced by a combination of top-down and bottom-up approaches, i.e., in which both top management and expert employees can take initiatives.

Although human-centered MSs help to integrate new job profiles and skills and human operators play a central role in Industry 4.0, few works seem to exist in the literature that consider the human factor in decision problems at different time horizons. Among the few examples, Moreira et al. (2017) deal with worker heterogeneity in ALBP. In the same context, Puthenveetil et al. (2015) use Motion Capture (MoCap) technology to have a real-time ergonomic assessment, and AR technology to assist the worker in its postures at workstations.

2.3.2.3 Adaptable MS

The VUCA world leads to change the modeling of MSs. An analysis of the literature reveals a greater effort is made at the strategic/tactical level to develop and use modeling methods to consider uncertainty. At the same time, at the operational/real-time level, a greater attention is put on methods capable of dealing with dynamic aspects (see Subsection 2.4.1.1).

As the behavior of MSs become more dynamic, RMSs need to evolve and integrate a decentralized control approach. In addition to the six characteristics of RMSs, Industry 4.0 requires the addition of a new characteristic, *Adaptability*, which allows for self-configuration and auto-adaptation using decentralized control.

Definition 2.2 *Adaptability is the ability of each system component to react to unexpected changes, act as an autonomous decision-maker, and cooperate with the other system components to optimize decision-making.*

Some works exist focusing on the adaptability characteristic of MS. For instance, Fragapane et al. (2020) propose a new analytical model considering the throughput and cost for comparing traditional production lines using conveyors as a system for intralogistics tasks and autonomous mobile robots (AMR)-based production networks

where AMR are used as transport systems. The latter uses AI algorithms to navigate through dynamic environments and provide optimized routing, enabling the conversion of traditional production lines into flexible production networks that distribute the material to different workstations and increase overall system flexibility. Nagy et al. (2020) exploit the multilayer network-based representation of production lines to represent connections between operators, skills, equipment, activities, and precedence constraints.

Changes occurred also in the way decision problems are modeled to make an MS more adaptable against several sources of uncertainty, such as variability, ambiguity, and imprecision.

Subsection 3.7.1 gives more details on the impact of adaptable MS on decision problems.

2.3.3 Transversal Evolution

Today, the VUCA world compels companies to resort to real-time monitoring. Besides, alternative control approaches should be adopted to overcome the limits of centralized approaches. This is indeed possible because decision-makers become capable of much more structured decision processes than before, thanks to the availability of massive amounts of field data for in-depth analysis.

Industrial companies define key performance indicators (KPIs) to monitor and control the performance of their MSs. KPIs help decision makers to assess the impacts of decisions on MS performance either before or after implementing decisions, which corresponds to proactive or myopic approaches, respectively.

In the proactive approach, decision makers incorporate KPIs in DSS to assess the behavior of the model the DSS is based on (see Subsection 1.4.2). The goal is to anticipate the impacts of decisions before implementing them in the real world. In contrast, in the myopic approach, data are collected from the plant, analyzed, then aggregated in order to compute the KPIs and identify the impacts of decisions. As a consequence, classical KPIs used to evaluate MS performance must be adapted to the context of Industry 4.0.

In the following, we present the main tranformation drivers of the transversal evolution, namely: the real-time monitoring, and decentralized control approach, and the sustainability-oriented KPIs in MSs.

2.3.3.1 Real-Time Monitoring

Rapid changes in customer needs, disruptive events, and short PLC call for the development of new techniques for real-time analysis that can deal with these challenges. Several authors have been interested in developing the appropriate algorithms to react quickly when such events occur by proposing a new system reconfiguration (Manceaux et al., 2016).

Smart objects possess the ability to monitor themselves. They are aware of their status and can ask for maintenance or help to fix the problem if they detect any defect. This process has been dramatically improved by the development of CPS (see Subsection 1.2.3.3) since the cyber system represents a reference for the physical one and facilitates system diagnosis. Furthermore, historical data records such as telemetry data, traceability product and processes, are directly accessible via cloud storage. Therefore, we could use these records as a comparison or a tool to upgrade the parameters of the cyber system.

Real-time monitoring offers many advantages, such as improving efficiency by early detecting anomalies in the process, extending components life by using them in their optimal conditions, and enabling decision-making by giving the actual state of the system and proposing alternative options accordingly. To fully benefit from these advantages, the decision-making process in MS involving real-time monitoring needs to be adjusted to the appropriate frequency to prevent MS instability or blocking states. For instance, a resource that must receive confirmation from an orchestrator before operating will not perform it if an event triggers the confirmation review just before the operation execution, leading to MS blocking. Therefore, we still need a freezer zone in production scheduling where decisions cannot be altered for instability reasons.

2.3.3.2 Decentralized Control Approach

Centralized control approaches are no longer suitable with the architecture of Industry 4.0-oriented systems (c.f. Subsection 1.2.2), which consists of a collaboration between components to monitor and deal with the complexity of a smart factory. Continuous change of process requires a flexible approach to allow to achieve reconfigurability with minimum effort. A decentralized control approach enables this flexibility and facilitate the dispatching of operations on different nodes to deal with a smart factory complexity (Morgan et al., 2021).

In addition, decentralized solutions provide better control of the shop-floor called distributed control, hence enabling self-configurable and auto-adaptive MSs. This type of control leads to self-organization, i.e., capable of adapting autonomously (Liu and Xu, 2016). However, this can happen in a decentralized control approach only if there is consistency in the exchange of data. Each entity in the system needs to adapt and reorganize its behavior based on the overall state of the MS, which requires communication with other entities. Even though the approach allows for local decision-making, entities still need to exchange data with external entities, and this exchange must be consistent to enable autonomous reconfiguration. The *line information system architecture* Theorin et al. (2017) is an example of flexible and scalable architecture that supports the development of distributed control .

Multi-agent systems (MASs) solutions can also help the development of decentralized

control approaches. Within industry 4.0, IoT is an appropriate technology to implement MASs. Each connected component, such as a product, machine, device, or AGV, can act as an agent. Each agent has its behavior, and can regularly interact with the other agents of the network and can collaborate to make a local decision before it is broadcast in the system.

For example, Schuhmacher and Hummel (2016) develop a proof of concept of Self-Execution System (SES) architecture, which is a framework based on an event-oriented concept. The SES aims to support future production scenarios using decentralized control approaches and a bidirectional exchange of information between the physical factory and the digital planning environment. Two main elements compose the SES: the system-related nodes, and the scenario-specific nodes, with nodes playing the role of agents. These system-related nodes are necessary for the system to function properly, and they are typically independent of use cases that the system might be designed to address, while scenario-specific nodes are those that are tailored to a particular use case or scenario. The prioritization of production orders is an example of services offered by system-related nodes, while scenario-specific nodes will cover services offered by the resources such as workstations, machines, and workers. The authors claim that the main advantage of SES in the context of a manufacturing execution system (MES) is its open architecture and the easy extensibility of the operating range. As a result, we could add agents that cover different functionalities with minimum effort. This is possible due to the event channel that plays the role of enterprise service bus decoupling the nodes and making the implementation of the modularity principle easy.

Before, presenting the sustainability-oriented KPIs in MSs. First, we briefly review the concept of KPI, and provide an overview of classical KPIs.

2.3.3.3 Concept of KPI

ISO (2014a) defines a KPI as "*a quantifiable level of achieving a critical objective*". Similarly, Ahmad and Dhafr (2002) considers it a value that indicates performance against an internal or external target.

The main role of KPIs is to understand and capture changes to drive the improvement of MS performance. Decision makers use them to reflect the current state of MS or identify decision impacts on MS from a specific perspective. For instance, from a business perspective, profitability indicators are used to judge the financial health of the enterprise or an ROI as a KPI to evaluate an investment decision. From a manufacturing perspective, an overall equipment effectiveness (OEE) indicator reflects the global performance of MS, while productivity allows evaluating a production scheduling decision.

Decision makers choose a set of KPIs for each MS activity and determine a suitable frequency of a measure depending on the expected objectives. Commonly, to represent

values of KPIs, dashboards are used that facilitate the MS monitoring and thus simplify the decision making.

With Industry 4.0, the cyber level of 5C architecture (see Subsection 1.2.3.3) is used to transform huge amounts of collected data into knowledge by performing detailed cross-analysis and computing KPIs. Nevertheless, a performance measurement system (PMS) is required to define and structure these KPIs at the cyber level. According to Neely et al. (1995), a PMS is a set of metrics able to quantify the efficiency and effectiveness of manufacturing operations.

2.3.3.4 Classical KPIs in MSs

Commonly, in MSs, five KPIs classes are defined: cost, time, quality, flexibility, and productivity (Hon, 2005). The factors used to calculate productivity are closely tied to labor time. Therefore, it is feasible to include productivity in the time class. Moreover, Chryssolouris (1996) evaluates the flexibility by the expected cost to adapt to possible changes in the MS.

In the literature, several frameworks are proposed to structure KPIs and build PMS for MSs. Among them, we cite the balanced scorecard (Kaplan, 2009), and the activity-based costing (Gosselin, 2006). Nevertheless, within Industry 4.0, companies need new frameworks that focus on data value and integrate a PMS as a component of their DSS.

A growing research stream concerns how to exploit data to standardize the computation of KPIs. To this end, ISO (2014b) defines 34 standard KPIs for MS (Ferrer et al., 2018). Moreover, recent articles attempt to structure KPIs. For example, Ante et al. (2018) proposed a hierarchical structure of KPIs to measure performance from strategic, tactical, and operational perspectives. Despite these efforts, the focus is still on the three main classes cited above (cost, time, quality), which are economic MS performance-oriented KPIs. Nevertheless, from the point of view of Industry 4.0, connectivity and sustainability dimensions impose taking into account other aspects for the assessment of MS performances. Therefore, new and more appropriate KPIs should emerge.

2.3.3.5 Sustainability-Oriented KPIs and Agility in MSs

In this subsection, we will deal with a part of transversal evolution that consists of using new KPIs for industry 4.0. After reviewing the work of the last five years for dealing with decision problems, we remark that many authors increasingly integrate two aspects as main criteria. First, the ergonomic aspect, especially to include humans at the design phase of MS. Second, the sustainability aspect, i.e., the environmental, economic, and social impacts of the MS. Finally, Agility is another aspect that can be considered in the

supply chain, even though rarely investigated in MS as a KPI. With the leagile strategy development, MS agility will reflect the reactivity of MS within Industry 4.0.

In what follows, we present recent works dealing with Industry 4.0-oriented KPIs (which we will be refering to as KPIs 4.0 from now on) to solve decision problems related to the three MS activities.

The ergonomic aspect plays a central role in the smart factory due to the increased human-machine collaboration, especially with the advent of cobot and AR technologies. Several authors highlight the necessity to integrate ergonomics as a criterion to design and manage MSs within Industry 4.0. Chengalur (2004) distinguish three ergonomic assessment methods: quantitative, semi-quantitative, and qualitative.

Abdous (2019) is an example of work that considers the ergonomic risk quantitatively by including fatigue and recovery in their model to design an MS, more precisely in the solving of an ALBP. The goal is to optimize cost and productivity. In addition to technical criteria, Bougrine et al. (2017) consider other ergonomic indicators such as posture, load, and working conditions to choose the best configuration of a Reconfigurable Manufacturing System (RMS). Shorter product life cycles and the emergence of mass customization have pushed companies to deal with product varieties. Therefore, they are looking to reduce or at least maintain adequate reconfiguration times. For this purpose, recently, Bortolini et al. (2021) combine technical and ergonomic objectives in RMS to reduce the repetitive movement of operators. Besides, Battini et al. (2016) prove that including ergonomics in the supply activity to determine the number of assembly operators and supermarket operators improves MS productivity. The picking process in the storage activity is the most concerned by ergonomic aspects, mainly when dealing with the picker-to-parts picking system. In this regard, Calzavara et al. (2017) propose economic models based on the order picking time and an ergonomic model based on the energy expenditure estimation concept to evaluate different rack layouts.

Concerning the management of MSs, Kovalev et al. (2019) include an individual ergonomic risk as criteria to deal with the scheduling problem and propose the assignment of workers to jobs over time periods. Similarly, Diefenbach et al. (2020) consider the ergonomic tow train loading problem for assigning bins to wagons and then bins to slots. Each wagon carries a shelf with racks, and each rack is divided into slots, where each one can store several bins. Battini et al. (2016) use the rest allowance concept to consider the ergonomic conditions in the order picking system. Authors use the OWAS (Ovako Working posture Analyzing System) method to assess ergonomic suitability because they consider it more appropriate in the storage activity. Further, to organize the planning of order picking operations, i.e., making the sequencing and routing decisions and scheduling pallet rotation tours, Glock et al. (2019) developed an integrated model, in which ergonomic evaluation is considered at each workstation. All these works show that ergonomic considerations impact different decisions of the three

activities of MSs at different decision levels. For more details about the physical, ergonomic risks at assembly lines, the reader can refer to the survey of Otto and Battaïa (2017).

As for sustainability, companies are becoming more and more aware of the importance of considering the environmental, social, and energy dimensions when designing and managing their MS. Today, it becomes necessary for the smart MS to converge towards sustainable manufacturing, as underlined in Subsection 1.2.2.

At the design phase, Touzout and Benyoucef (2018) deal with the process planning problem in RMSs to optimize three criteria: production cost, completion time, and greenhouse gas emission. It becomes more common within industry 4.0 to use AGV or autonomous mobile robots (AMR) for supply activity. However, introducing such vehicles requires taking their energy consumption into account. For instance, Zhou and Shen (2018) propose to consider AGVs loading weight and travel distance in order to compute the energy consumption. Furthermore, the authors tackle the energy-efficient part feeding problem in the case of mixed-model assembly lines. They include two tactical decisions already described in the supply activity: routing and scheduling problems. Recently, Bartolini et al. (2019) state that the most studied objectives to consider sustainability in the warehouse are energy saving, environmental impact of the warehouse building, and green warehouse management. The authors provide a review to highlight the importance of taking carbon footprint into consideration in different warehouse processes at the design and planning phases. Therefore, it is necessary to implement new energy-saving technologies or install safety equipment. Malinowska et al. (2018) propose a model to assess the level of the sustainable warehouse.

At the management phase, Ghanei and AlGeddawy (2016) consider a scheduling problem and use an optimization model to derive the system configuration and job sequencing according to variations of energy prices throughout the day. Briand et al. (2018) tackle the loading problem, again considering the energy dimension. They include the load, the number of stops and the distance traveled to provide an energy-efficient supply plan. Finally, Ene et al. (2016) consider the case of manual warehouses and propose a bi-objective model to integrate the batching and routing problems and minimize travel time and energy consumption.

Similarly to the ergonomic aspect, we could deduce that sustainability is becoming critical for the three activities at different phases, and that among the aspects involved in sustainability, the energy dimension is the most investigated.

In addition to the sustainability KPIs, defining a new KPI to measure the MS adaptation and proactivity ability is required to face the dynamic changes occurring in Industry 4.0. For this reason, we introduce the agility KPI. The concept of agility in MSs dates back to the 90s. Ramasesh et al. (2001) assert that agility is a multi-dimensional attribute of MS. They define it "*as the capability of a manufacturing system to provide an effective response*

to unanticipated changes". Foundational social and life science theories identify the five dimensions that define the agility of a supply chain: alertness, accessibility, decisiveness, swiftness, and flexibility (Gligor et al., 2013). Below, we use these five dimensions to propose an updated definition of MS agility:

Definition 2.3 *The MS agility is the ability to detect changes quickly, access relevant data rapidly, use available information to make decisions, and then implement these decisions on the MS to modify its configuration to the extent needed.*

MS agility concerns all MS activities. For instance, in production activity, an agile MS is capable of changing the line configuration quickly to meet customer needs in terms of volume and variety or acting before failures occur. This is also true, in the supply activity, for the capability of quickly changing fleet size from period to period without additional cost. Finally, in the storage activity, adapting operation strategies automatically to the production sequencing & scheduling can be seen as an example of MS agility, i.e., modifying the routing strategy depending on the components required for the production sequence, or choosing the storage strategy according to the production schedule that provides the time of stay of components. Similarly, the picking strategy is selected so as to ensure synchronization with the production throughput.

For all these reasons, the agility of an MS is likely to be among the KPIs that will stand out with Industry 4.0.

So far, we have identified the technical, organizational and transversal evolutions that occurred in industry 4.0 from a decision-making perspective. In the following, we focus on the evolutions of decision problems in industry 4.0.

2.4 Evolution of Decision Problems in Manufacturing Systems in Industry 4.0

In Section 2.2, we asserted that decisions could be made either at the strategical, tactical, or operational level. In Subsection 2.4.1, we will study how the the identified tranformation drivers will impact the time horizon of decisions.

Moreover, in Section 1.2, we distinguished three MS types: DMS, FMS, and RMS. In Subsection 2.4.2, and according to the drivers cited above, we will analyze whether these MS types are still compatible with the industry 4.0 scenario, and if not, how must they be adapted to meet the industry 4.0 requirements.

2.4.1 Evolution of Time Horizon of Decision Problems in MS

Many factors contribute to determine the time horizon of a decision problem: the frequency of decision-making, the business sector of company, the nature of the decision, i.e., whether it is related to design, planning, or control. Most of the time, design

decisions involve investments, and they may impact the strategy of company, while planning decisions impact the organization, and control decisions act on daily management.

The main goal of studying the evolution of the time horizon of decisions is to investigate whether the identified drivers are suitable or not to support these evolutions.

Based on our literature review and industry field analysis, it can be observed that some strategic decisions remain strategic, while the criteria for addressing these decisions evolve. For example, we take into account criteria such as the reconfiguration time or ergonomics at the design phase. Another remark is that the frontier between strategic and tactical decisions is becoming increasingly unclear since Industry 4.0 accelerates strategic decisions, and most tactical decisions become operational. Lastly, real time is emerging due to the evolution of business models and enabling technologies. Hence, some operational decisions become real-time decisions. Real time decisions is the first class of decision problems that we consider in the following. Then, we will discuss the evolution of the time horizon of each decision problem.

2.4.1.1 Real-Time Horizon

The new context of Industry 4.0 has led to taking into account a new time horizon, called real time. The name refers to the frequency and the short interval time available to make a decision.

According to Csalódi et al. (2021), the urge towards real-time capability is one of the major effects of Industry 4.0 on optimization. Uncertainty aspect, dynamic data, and disruption events in MSs are among the reasons for this evolution. For example, the solution of the decision problem at a given time may be no longer only a very short laps of time later. Furthermore, it is not always possible to change the solution implemented when uncertain events occur, for instance, due to critical computation time.

Real-time or near real-time decisions have become possible thanks to the development of technologies that allow to capture the information required to make a desicion in real-time. Based on acquired data, machine learning algorithms and techniques allow to rapidly detect anomalies and formulate a suitable decisions to react to them. IoT, especially RFID technology and wireless communications, are mainly used to capture machine data in real-time. CC technology is also used to improve real-time interaction between MS and decision makers, e.g., by allowing the personalization of customer orders in near real-time. In the following, we present examples of decision problems considered at the real-time horizon.

In production activity, Wang and Jiang (2016) propose a rescheduling decision mechanism by using a manifold learning (a Machine Learning algorithm) to choose the strategy, policy, and method selection of rescheduling. Khodke and Bhongade (2013)

call this scheduling typology predictive-reactive scheduling, which consists in reviewing the schedule following real-time events.

Klenk and Galka (2019) provide an example of a real-time decision in the supply activity. The authors remind that the loading problem often considers fixed routes and predetermined planning based on the fixed time interval between consecutive executions of a specific route. With Industry 4.0, decision makers must deal with variations in the number of orders caused by changes in production planning or unequal container lot sizes. Therefore, the authors propose to adapt part of the feeding configuration in real-time, i.e., to solve the routing and scheduling problems for different load scenarios. For this purpose, they use a Discrete Event Simulation (DES) approach to assess the performance of different strategies. In the same context, Chibani et al. (2018) focus on agile supply chain procurements is interesting to investigate due to the high similitude of the problem of determining the suppliers and ordered quantities over time, with the scheduling problem in the supply activity. The authors propose a dynamic optimization approach, specifically a dynamic genetic algorithm, capable of quickly reacting to frequent changes. They adopt memory approaches and introduce diversity when changes occur to ensure a diversified population of solutions and enable the algorithm to adapt to changing objectives.

As for the storage activity, Cao et al. (2017) use the RFID technology to solve the order scheduling problem, which includes the order batching, and routing and sequencing decisions. Furthermore, since an accurate tracking of the warehouse is made possible by real-time data collection, they deal with rescheduling problems in case of deviation between an expected scheduling and the actual execution. They propose a genetic algorithm to solve these two problems and present a case study designed in their warehouse laboratory. Tu et al. (2020) develop a CPS-based pick-and-pass system (PKPS) and uses a heuristic multi-objective genetic algorithm to solve the SLAP. As stated by the authors, a promising perspective is represented by the development of a real CPS-based PKPS that acquires real-time SKU data.

2.4.1.2 Evolution of Time Horizon of Some Decisions

Table 2.2 highlights some time horizon evolutions of decision problems belonging to different activities. The table is based on our analysis of trends and evolutions mentioned in Section 2.3.

In Table 2.2, the first column presents the decision problem among those already identified in an MS. The second column denotes the initial time horizon, while the third column describes the expected trend of the time horizon evolution, which is discussed in the references cited in the last columns. The term “expected” is used as it is not yet widely adopted in the literature. The main objective of the Table 2.2 is to demonstrate the current status of changes that have occurred, or not, concerning the time horizon of decisions and to highlight emerging works in this area. The abbreviations S, T, O, and

Table 2.2: Evolution of time horizon of some decisions in MS in the context of Industry 4.0.

Decision problem	Initial time horizon	Expected new time horizon	References showing this trend
Production Strategy	S	S-T	(Steier et al., 2021)
Capacity Investment	S	S-T	(Wu et al., 2021)
Aggregate Planning	T	O	(Bueno et al., 2020; Yu et al., 2015)
Capacity Allocation	O	O	(Ghasemi et al., 2020; Wang et al., 2021)
Choice of Type of Assembly Line	S	S	(Cohen et al., 2019; Pattanaik and Jena, 2020)
Line Balancing	S	S-T	(Abdous, 2019; Campana et al., 2021)
Equipment Selection	S	S-T	(Mura and Dini, 2019)
Facility Layout	S	S-T	(Peron et al., 2020)
Line Layout	T	O	(Hosseini-Nasab et al., 2017)
Line Rebalancing	T	O	(Belassiria et al., 2018)
Line Relayout	T	O	(Song et al., 2021)
Maintenance strategy selection	S	ST	(Silvestri et al., 2020)
Maintenance load forecasting	T	O	(Posadas et al., 2020)
Maintenance capacity planning	T	O	(Sala et al., 2021)
Maintenance scheduling	O	O	(Li et al., 2020)
Maintenance control	O	RT	(Mitra et al., 2021)
Sequencing & Scheduling	O	RT	(Grassi et al., 2020; Shiue et al., 2018)
Lot Sizing & Scheduling	T-O	O-RT	(Dolgui et al., 2018)
Sizing	S	S	(Fragapane et al., 2020)
LPFS	S	S	(Fathi et al., 2020)
Routing	T	O	(Flores-García et al., 2021)
Scheduling	T	O	(Zhou and Zhu, 2020)
Line feeding Policy	T	O	(Boysen et al., 2015; Neumann et al., 2021)
Loading	O	RT	(Klenk and Galka, 2019)
Inventory Policy	T	O	(Lee, 2020; Zheng et al., 2020)
Inventory Planning and Control	O	RT	(Yuan, 2020)
Storage Equipment Selection	S	S	(Bartolini et al., 2019; Custodio and Machado, 2019)
Material Handling Selection	S	S	(Horňáková et al., 2019)
Dimensioning and sizing	S	S-T	(Shi et al., 2017)
Storage Layout	S	T	(Raghuram and Singh, 2020)
Block Staking Problem	S	S-T	(Venkitasubramony and Adil, 2018)
Configuration of AS/RS	S	S	(Hamzaoui et al., 2019; Schenone et al., 2020)
Zoning size problem	T	O	(Silva et al., 2021)
Zoning assignment problem	O	O	(Fragapane et al., 2021)
Storage Strategy	T	O	(Alfathi et al., 2019)
Picking Strategy	T	O	(Winkelhaus et al., 2021)
Routing policies	T	O	(Elbert et al., 2017; Lee et al., 2019; Montanari et al., 2021)
SLAP	T-O	O-RT	(Keung et al., 2021; Tu et al., 2020)
Routing & Sequencing	O	RT	(Cao et al., 2017; Olmos et al., 2021)
Batching	O	RT	(Leung et al., 2020)
Workforce Assignment	O	RT	(Kong et al., 2018)

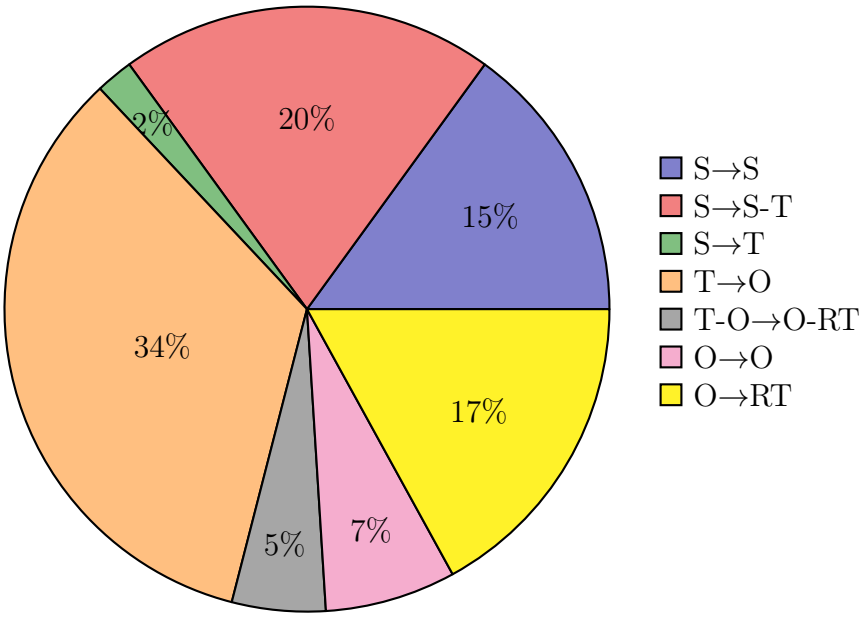

Figure 2.1: Changes of the time horizon of decision problems from the old scenario into Industry 4.0 scenario.

RT respectively stand for Strategic, Tactical, Operational, and Real-Time decision.

Figure 2.1 summarizes the evolutions of time horizon of decisions showed in Section 2.3 to seven movements. In it, symbols S, T, O, and RT have the same meaning than in Section 2.3. For instance, S-T or T-O mean Strategical-Tactical or Tactical-Operational level, respectively. More than a third of decisions evolve from tactical to operational level. Besides, 20% of strategical decisions become strategical-tactical decisions, while 17% of decisions evolve from operational to real-time level. These three cases cover more than 70% of the evolution of decision time horizons, clearly showing the trend of how decisions are evolving and proving that the leagile strategy is required because the time for making a decision becomes shorter than before, and reactivity must be targeted for all decisions at different time horizon.

In Industry 4.0, some strategic decisions remain at the strategical level due to the lack of standardization in Industry 4.0-oriented equipment. Nowadays, it has become uncommon to purchase standard equipment as suggested in vendor catalogs, the industrial companies are prioritizing personalized equipment tailored to their specific needs. Further, they ask for unique capabilities, such as the ability to generate equipment data (e.g., energy consumption, health indicators), support for communication protocols (e.g., ProfiNet, OPC-UA, MQTT), and modularity of design to enable scalability. Unfortunately, obtaining such customized equipment can be a long and arduous process, taking anywhere from six months to a year to go from specification to implementation and validation. This prolonged lead time means that strategic

decisions, including the choice of assembly line types, sizing, or storage equipment selection, often extend beyond a year. As a result, these decisions remain at the strategical level.

The only case of time horizon shifting from strategic to tactical is the storage layout, due to the new algorithms and techniques proposed to improve the indoor positioning systems (see for instance, (Bencak et al., 2022)).

Evolutions from tactical-operational to operational-to-real-time decisions are mainly due to advances in techniques and algorithms. For instance, Dolgui et al. (2018) discuss the scheduling in production within Industry 4.0 and distinguish three strategies: purely reactive, predictive-reactive, and robust-predictive-reactive. It means, the scheduling can be considered as near real-time decision. With the emergence of Demand-Driven Material Requirement Planning (DDMRP) method, lot-sizing decisions move from tactical to operational level thanks to the possibility to adapt the planning day by day, thus specifying the number of manufacturing orders and adjusting the quantity to produce in each order. Therefore, the lot-sizing and scheduling decision problem is considered as operational-to-real-time decision. Similarly, using the reinforcement learning to deal with dynamic SLAP allow to consider SLAP as an operational-real-time decision.

According to Table 2.2, decisions remaining at the operational level are the capacity allocation problem, the maintenance scheduling problem, and the zoning assignment problem (Fragapane et al., 2021; Ghasemi et al., 2020; Li et al., 2020).

When dealing with capacity allocation problem, Industry 4.0 environment imposes to take into account constraints such as resource capability (qualification of resource for different recipes), and resource dedication (using specific resources for some tasks). For example, Ghasemi et al. (2020) propose a new formulation to integrate such constraints in capacity allocation problem in the semiconductor manufacturing at operational level. Li et al. (2020) deal with the integration of maintenance scheduling and the production planning problems within the framework of intelligent manufacturing. They retrieve production data from MES and data related to equipment maintenance from another system, before deciding preventive maintenance (PM) interval and duration, and optimize the schedule of the PM tasks on a daily basis. For what concerns the zoning assignment problem, Fragapane et al. (2021) discuss the impact of new technologies on planning and control decision for automate mobile robots (AMRs). They state that the use of AMRs in intralogistic leads to define zones in which they can operate autonomously with high flexibility , since we can change them on a daily or weekly basis. Furthermore, the authors specify that the personalization of tasks such as picking items or human collaboration can be added and assigned on a short-term basis.

2.4.2 Evolution of Manufacturing Systems Type

DMS characteristics do not match with the Industry 4.0 challenges such as short PLC and new business models. Moreover, our literature review reveals that even research in DMS is becoming very scarce. That is why in our vision of decision making, we disregard DMSs in Industry 4.0 scenario.

On the other hand, FMSs meant for multiple preplanned products, which means that they are not designed to adapt quickly to changes in production demands or changes in the manufacturing process. Further, they lack connectivity and interoperability with other systems. Therefore, FMSs cannot keep up with the speed and flexibility needed by Industry 4.0.

Although RMSs have advantages over both DMSs and FMSs, there is still room for improvement, particularly in terms of decentralized control approaches required to reduce process complexity in Industry 4.0. The control system is crucial for coordinating production processes, monitoring industrial IT systems, and optimizing processes to meet customer requirements.

The development of an RMS controller and software as a CPS is required to easily integrate the adaptability characteristic (see Subsection 2.3.2.3) in RMSs. More information can be found in Subsection 1.2.3.3 and Subsection 1.2.3.4.

To conclude this analysis on the evolution of MS types, Zhong et al. (2017) consider the intelligent manufacturing system the next generation of MS, which optimizes production by using advanced ICTs and technologies and improves the PLC from design to production and management. Cloud manufacturing is another impending evolution of MSs as said in Subsection 1.2.3.1. Cloud computing and service-oriented technologies will act as the support of this new production paradigm, and enabling end-users to use the manufacturing resources as services through virtual platforms (Xu, 2012).

2.5 Conclusion

In this chapter, we first identified the existing decision problems in MSs for each of the three main activities, namely, Production, Supply, and Storage.

Next, we provided the main drivers of the transformation from the old to the Industry 4.0 scenario. This transformation showed the enterprise's perception change in Industry 4.0, hence, the necessity to change the way of enterprise modeling. In other words, we need to analyze the impact of the main transformation drivers thoroughly to summarize Industry 4.0 requirements from the point of view of decision making. Further, we analyzed the evolution of MS performance in Industry 4.0. Discussions about classical and new KPIs were carried out to highlight a fundamental change in designing and managing MSs within Industry 4.0.

Specifically, the leagile strategy urges companies to be more agile in their modeling to enable a proactive approach rather than myopic, while the decentralization control approach imposes modularity as a principle of modeling i.e., each MS element must act autonomously and interact with other elements. Moreover, the standardization of data exchange forces the consideration of the interoperability issues at earlier modeling stages. Furthermore, real-time monitoring requires adopting data-driven models. Hence, data must be collected, analyzed, processed, and then used to generate models rapidly.

On the other hand, the assessment of MS performances lead to the definition of new KPIs and their effective integration in the modeling process. However, automatic generation of models is not expected to be achieved in a short future, due to the need of integrating human skills and judgement, hence the need of MS to be still human-centered. Instead, the focus must be on consistency in ensuring the correct integration of models. Finally, the new modeling techniques prove that optimization, simulation, ML techniques, and their coupling are the most promising techniques to face the challenges of Industry 4.0. This observation is logical with the evolution of skills (mentioned in Subsection 2.2.2.2).

Finally, we investigated the evolution of decision making within Industry 4.0 w.r.t. the time horizon, and MS type. From the analysis of the results of Section 2.4, it becomes clear that we need an EM methodology that guides the modeling process in order to be able to derive DSS-oriented Industry 4.0. However, before doing that, we note the necessity of proposing a taxonomy of decisions in MS and designing new tools that promote the development of integrated models. This goal will be the objective of the next chapter.

Chapter 3

Modeling of Decision Problems in Manufacturing Systems and their Interactions

Résumé

Ce chapitre aborde la prise de décision dans les systèmes de production et son évolution à l'ère de l'Industrie 4.0. Il propose une nouvelle approche pour étudier les problèmes de décision et leurs interactions dans les systèmes de production. Ce chapitre se divise en trois étapes : la première consiste à proposer une taxonomie des problèmes de décision identifiés dans les systèmes de production en fonction de quatre facteurs: l'horizon temporel, la famille du problème, la branche du problème et le type d'activité. Ensuite, il introduit le concept des propriétés de décision (contraintes, entrées, hypothèses, sorties et critères), qui sont utilisées pour décrire les décisions de manière générique, adaptée aux deux scénarios, celui avant la révolution et celui de l'Industrie 4.0. La deuxième étape concerne la modélisation des interactions entre les décisions, qui permet de comprendre comment les décisions s'impactent mutuellement. Trois types d'interactions sont identifiés : corrélées, séquentielles et simultanées.Une cartographie des décisions est ensuite proposée comme outil graphique pour résumer la taxonomie et les interactions des problèmes de décision. Cet outil peut aider les chercheurs et les praticiens à identifier les synergies entre les problèmes de décision appartenant à différentes activités et à développer de nouveaux modèles intégrés et robustes. La troisième étape explore l'impact des principaux leviers de transformation de l'Industrie 4.0 (identifiés en chapitre 2) sur l'évolution des décisions dans les systèmes de production. Cette analyse met en évidence la nécessité de changer la manière de modéliser l'entreprise pour répondre aux exigences de l'Industrie 4.0 et soulève des questions concernant l'évolution des méthodologies de modélisation d'entreprise et la nécessité d'intégrer de nouvelles vues pour couvrir tous les aspects de l'entreprise. Ces questions seront traitées dans le prochain chapitre.

Contents

3.1 Introduction 96
3.2 A Three-Step Procedure to Study Decision Making in Manufacturing Systems 98
3.2.1 Basic Concepts 98
3.2.2 Conceptual Model of the Function View of an MS . . 101
3.2.3 Introduction of a Three-Step Procedure to Study Decisions in MSs 102
3.3 Decision Identification and Classification 102
3.4 Decision Definition 103
3.4.1 Conceptual Model of Decision View of an MS 103
3.4.2 Conceptual Model of Information View of an MS . . . 105
3.4.3 Conceptual Model of Performance View of an MS . . . 106
3.4.4 Properties of Decision Problems 108
3.5 Decision Interactions 108
3.5.1 Some Decision Interactions Linked to Plant Activities 110
3.5.2 Types of Decision Interactions 113
3.5.3 Rules to Study Decision Interactions 114
3.5.4 MS Type-Dependent Decision Interactions 117
3.6 A Graphical Tool to Represent Decisions in MSs and How they Interact with Each Other 119
3.7 Using the Cartography Tool to Represent the Evolution of Decision Problems and Interactions among them in the Industry 4.0 Context 121
3.7.1 Evolution in the Time Horizon of Decisions 121
3.7.2 Evolution of Decision Interactions 133
3.8 Conclusion 136

3.1 Introduction

MS results from successive decisions made at different levels and times of the MS life cycle, covering all plant activities. Understanding MS behavior requires analyzing its business processes, the decisions involved in it, and how its performances are impacted.

However, as we analyze the impact of Industry 4.0 on Enterprise Modeling (EM), it is becoming increasingly clear that the process view may no longer be suitable. First, since the complexity of processes is increasing, trying to use the process as a core model can lead to models including case-specific details of processes. Second, the modularity principle required by Industry 4.0 forces to change the focus from process to function satisfied by the process. Third, the technological, economic, and organizational transition caused by Industry 4.0 is ongoing, and attempts to obtain suitable EM frameworks are sparse and case-specific; therefore, no stable standardization of processes exists yet that is capable to adapt to the current, ever changing industrial scenario. As a consequence, it is necessary to change the mindset of designing an EM methodology. To this end, the work of Jørgensen (2009), provides some insights regarding enterprise modeling and states

that: "*Processes are better understood by focusing on the decisions to make, the issues to solve, and the results to produce, than on the administrative ordering of steps.*"

In classical EM frameworks, the organization view is used to study decisions or to consider the decision modeling in processes, i.e., as process steps. Here, we prefer to separate the decision view from the organization and process views, as MSs and smart MSs result from several decisions. Investigating these decisions in their own right allows for a more comprehensive understanding of the organization. Further, it helps to identify opportunities for automation and optimization of decision-making processes using Industry 4.0 technologies. Besides, within Industry 4.0, we highlited the emergence of new time horizon called "real time" that leads to real-time decision making. In this context, the decision view model helps companies to analyze and optimize their decision-making capabilities, ensuring that decisions are made in a timely and data-driven manner. In addition, companies need to use simulation for evaluating different decision strategies, because, a dynamic adaptability is required in Indsytry 4.0. Resorting to the decision view model facilitate to understand thouroughly decisions to make and promote the development of adaptive decision-making capabilities. Therefore, in our opinion, considering the model of the decision view as a core model is the best way to build an enterprise model adapted for the Industry 4.0 scenario. By doing so, we can better capture the complex decision-making processes that underpin MSs and design more adaptable, resilient, and efficient enterprise models.

Chapter 2 has presented the literature review of the decision making in MS and how it has evolved in the last decades. In this chapter, we propose to go in depth of each decision. To achieve this goal, we propose a new taxonomy which allow to identify the properties of each decision problem involved in plant activities, as well as interactions among decisions, so as to have a better understanding of the MS behavior and, therefore, of its performance. We also define conceptual models of some views from this chapter. First, because, the core model needs to be detailed and understanded well to be able to build a consistency enteprise model. Second, it is common to adapt a progressive approach when defining the conceptual models of views to make easy the understanding of concepts. Third, this chapter is dedicated for modeling decision-problems in MSs, and without providing the links with the conceptual models of direct related views will not give the common thread.

Section 3.2 provides the basic concepts of decision making in MSs. Then, we describe the framework of the decision process in MSs based on a novel three-step procedure, which we introduce here. This procedure covers decisions in MSs, from their definition to the study of the interactions between them. Section 3.3 proposes a taxonomy of identified decision problems in MS based on four factors: time horizon, problem family, problem branch, and activity type. Section 3.4 details each identified decision problem according to five properties, i.e., constraints, inputs, assumptions, outputs, and criteria

(CIAOC). Section 3.5 identifies the existing interaction types, investigate interactions among decisions, and defines a new formalism capable of modeling these interactions. Section 3.6 proposes a novel graphical tool to derive the cartography of decisions that illustrates decision problems, taxonomy, and interactions. Using this cartography, we illustrate an overview of changes that occurred within Industry 4.0 depicted in Section 2.4. Finally, Section 3.8 concludes this chapter. From the viewpoint of enterprise modeling, this chapter focus on function, decision, information, and performance views, as we will show in Sections 3.2 and 3.4.

3.2 A Three-Step Procedure to Study Decision Making in Manufacturing Systems

To study the decision view correctly, we propose in this section a three-step procedure. Before doing so, we define some concepts which represent the building elements of the procedure.

3.2.1 Basic Concepts

In addition to the concepts already defined in Section 1.4, namely, the decision problem and time horizon, we describe other basic concepts used in our procedure.

McCarthy and Menicou (2002) defines the *decision domain* as "*...any subset of an organization that has the necessary interest and importance to warrant a study*". Production planning and control, product development, or Workforce are examples of decision domains. Some of the most used enterprise modeling frameworks use the notion of decision domain as the highest level of aggregation when modeling the decision process of an MS. CIMOSA (Computer Integrated Manufacturing Open System architecture, Berio and Vernadat (2001)), or GIM (GRAI (Graph with Results and Actions Inter-related) Integrated Methodology), which builds a GRAI-grid for each decision domain, are examples Doumeingts et al. (1993). However, the GRAI-grid formalism has a major limitation, since it represents one decision domain at a time. This becomes a significant issue when a holistic vision is required, and all decision problems in an MS, or the possible relations among decision problems that belong to different domains, must be studied.

As stated in Subsection 1.3.5.1, the *activity* of a decision problem corresponds to the functional decomposition of an MS. The activity concept indicates a higher level of abstraction than the decision domain since it can group several domains. The decomposition of an MS into macro activities is, in our opinion, the most appropriate detail level to have a holistic view of decisions in MSs and of the interactions among them, although some decision problems in MSs concern decisions that belong to two activities.

A *problem family* consists of a set of decision problems related to one same aspect of the decision process of an MS. For example, *Line Configuration* is a step of the design of MSs, which encompasses three decision problems, namely *Choice of assembly line*, *Line Balancing*, and *Equipment Selection* (as we will see later in Section 2.4). The problem family concept is strictly related to the notion of the decision domain. For instance, the well-known decision domains are the inventory management domain (Guo et al., 2019), and the maintenance domain (Syan and Ramsoobag, 2019) that corresponds respectively to the inventory and the maintenance and process control problem families.

However, we decide to use in our classification the concept of problem family rather than decision domain since, in the literature, there is no standardization of existing decision domains in MS, while grouping decision problems per family is commonly used. On the one hand, Nestman and Windsor (1985) identify decision domains as either correlative, strategic, tactical, or operational, while, Yusuf et al. (1999) consider ten decision domains of an agile manufacturing enterprise: Integration, Competence, Team building, Technology, Quality, Change, Partnership, Market, Education, and Welfare. On the other hand, Plehn et al. (2012) distinguish three decision domains, namely, product design, machine design, and production planning and control.

In the literature, it is common to distinguish decision problems in *design* and *management* problems as the reader can see for example in the reviews of Gu et al. (2007, 2010); Negahban and Smith (2014). Based on this and the three aforementioned macro-activities, we introduce five *problem branches*: MS design, MS Management, MS Feeding, Warehouse Design, Warehouse Management. The first two cover the design and management of the production activity, and the same do the last two w.r.t. the storage activity; MS Feeding covers the entire supply activity since this distinction is less evident in the literature. The product design branch is excluded from our work. First, as indicated in Section 2.2, we assume that decisions about the choice of products and processes have already been made. Second, we decide to pay more attention to the manufacturing than the engineering part, even if the concurrent engineering shows clearly that the two parts interact during the whole life cycle of the MS.

Figure 3.1 shows the scope of our study in the form of a 2D matrix: vertical and horizontal axis represents time horizon and activities, respectively. Branches are represented by blue dotted boxes. By the end of this chapter, every MS problem decision concerned by the production, supply, or storage activity will be placed in this matrix.

Each decision has constraints, inputs, assumptions, outputs, and criteria, which we call *decision properties* and summarize with the acronym CIAOC. According to McCarthy and Menicou (2002), these properties define the microstructure of a decision.

Let us take the example of the Line Balancing problem, precisely the Simple Assembly Line Balancing-1 (SALBP-1), which consists of assigning production tasks to a minimal number of workstations from a straight-paced line to meet the customer demand

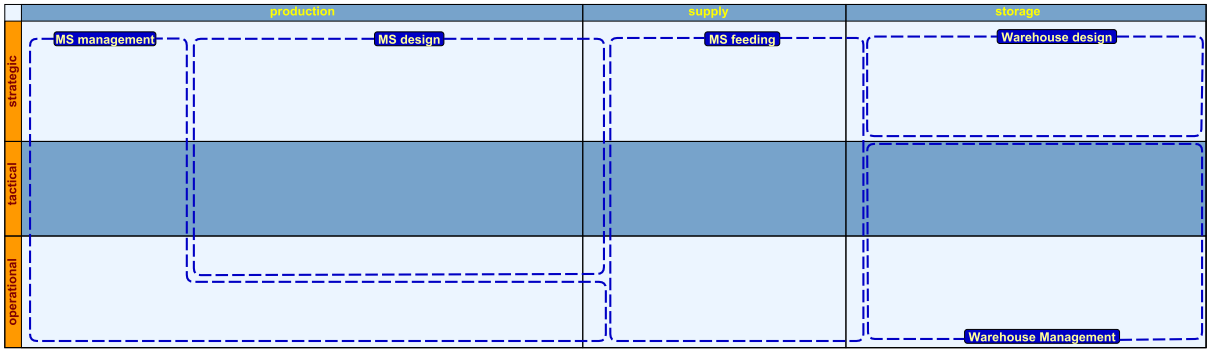

Figure 3.1: The scope of study covered in the manuscript.

(Battaïa and Dolgui, 2013). The constraints are what restrict the solution space, e.g., the precedence relationship among tasks or the takt time. Among the inputs, the task processing times are considered. The assumptions are the limits of validity of the study, e.g., considering a straight line of homogeneous workstations. The only output is the assignment of tasks to the workstations. Finally, the criterion is the objective to optimize, i.e., the number of workstations.

Decision properties of MS decisions lead to a modular vision of the decision process in MSs. Most of all, they highlight the *decision interactions*, i.e., whether and how decisions relate to each other, e.g., whether the output of a problem is part of the inputs or the assumptions of another.

We define decision interactions as binary relations, i.e., as relations between two decision problems A and B. We identify three main interaction types:

- *sequential interactions* in which, the sequence of the decision-making time is essential, i.e., decision A must be taken so that problem B can be stated and tackled.
- *correlated interactions* in which, decisions A and B can influence each other without there necessarily being a sequence between them. This occurs, for instance, when some assumptions of B can either be outputs of A or standalone assumptions. In the latter case, solving problem A is not strictly required in order to state and tackle problem B.
- *simultaneous interactions*, occuring when A and B can be tackled simultaneously using an integrated model, which gives the outputs of both A and B.

The notion of interaction is similar to the decision link of the GRAI-grid formalism of GIM. However, when using GRAI-grid, it is necessary to define a sequence between decisions, which is not consistent with the types mentioned above of decision interactions.

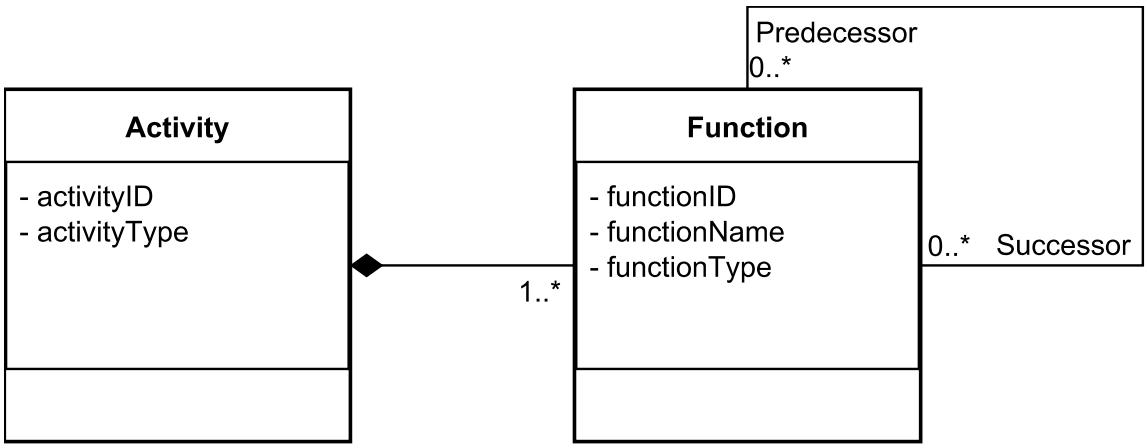

Figure 3.2: The conceptual model of the function view of an MS.

3.2.2 Conceptual Model of the Function View of an MS

As defined in Subsection 1.3.1, the view concept in enterprise modeling is crucial. We can focus on specific aspects by breaking down the enterprise into smaller, manageable views without overwhelming the entire model. This modular approach makes it easier to maintain and update the models as the enterprise evolves over time. It also enables scalability, allowing for adding or modifying views without affecting the entire model and causing interoperability issues. Each view can be used to identify and specify the requirements associated with its particular domain. Furthermore, it establishes a common language and framework that can be followed by all stakeholders involved in the modeling process, as well as ensures that the enterprise model is well-documented, understandable, and adaptable to changes in the business environment.

The definition of conceptual models ensures consistency and standardization across different views. A conceptual model is a UML class diagram (see Subsection 1.3.4.2). It allows defining a relationship between different abstract concepts, called classes in object-oriented modeling.

Figure 3.2 illustrates the conceptual model of the function view of an MS using two classes, Activity and Function. It defines what in UML is called a *composition* relationship between them, i.e., the Function object cannot exist separated from the Activity object. This model also indicates that each Activity could have several Functions (e.g., supply activity has a loading/unloading function, transport function). Moreover, an *association* relationship exists between two functions. This relationship shows that each Function can have any or several successors and/or predecessors (e.g., loading precedes the transport function, while unloading succeeds the transport function).

In the following, we define a new procedure to study decision problems in the three activities of the plant on which we decided to focus our work.

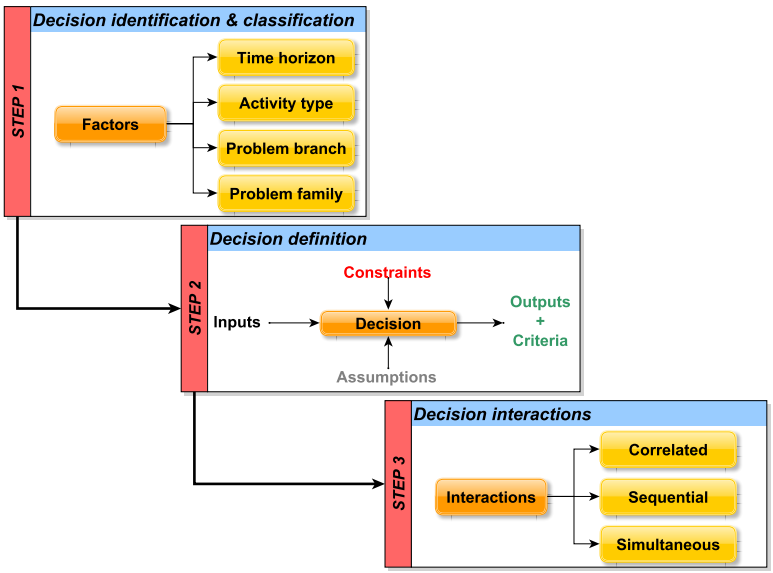

Figure 3.3: Basic structure elements of a novel three-step procedure to study decision process in MSs.

3.2.3 Introduction of a Three-Step Procedure to Study Decisions in MSs

The decision process of an MS is very complex. In order to simplify it, we introduce a straightforward three-step procedure (shown in Figure 3.3):

- the first step identifies the existing decision problems in MSs, before proposing a classification according to four factors: time horizon, problem family, problem branch, and activity.
- the second step specifies the CIAOC properties of decisions based on the MS features (see Subsection 3.4.2 for MS features definition).
- the third step studies the interactions between decisions based on the second step.

In the following, Section 3.3 goes in depth on step one; Section 3.4 describes in detail step two and provides the conceptual model of decision and information views; Section 3.5 focuses on step three of the procedure.

3.3 Decision Identification and Classification

The first step consists to identify and then classify the decision problems of MSs. The identification of the decision problems in MSs related to various plant activities has already been completed by Section 2.2. Now, we focus on their classification.

Table 3.1 provides a classification of existing decisions in MS according to the four above mentioned factors, namely, time horizon, problem family, problem branch, and activity. In the time horizon column, the abbreviations S, T, and O mean respectively Strategic, Tactical and Operational.

3.4 Decision Definition

Based on the short review of decision problems of Section 2.2, we propose to model an MS as an enterprise object represented according to the four concepts introduced by Labrousse and Bernard (2008): process, product, resource, and external effect (PPRE). We choose to refer to these concepts as classes, in order to be able to structure our decision definition. We decide to add two separate classes, namely cost and time (CT). Shah et al. (2014) consider cost and time as resources consumed by an activity, while Neely et al. (1995) consider them as two dimensions of performance measures. In this work, we adopt the latter viewpoint, as in our opinion it better fits our intention to study the performance of an MS object.

3.4.1 Conceptual Model of Decision View of an MS

In Subsection 3.2.1, we define what we call the CIAOC properties of decisions, i.e., the constraints, inputs, assumptions, outputs, and criteria. By decision constraints we mean the conditions defining whether a decision is feasible or not. Decision inputs are parameters of the system or some of its parts that are known before making a decision. Decision assumptions define the limits of validity of a decision. Decision outputs are the solutions yielded by the solving of the decision problem. Finally, decision criteria are the objectives we seek to optimize while making decisions.

The decision view captures the decision-making processes, rules, and criteria within the enterprise and enables simulation and evaluation of different decision strategies, facilitating agile and adaptive decision-making capabilities. By modeling this view, enterprises can analyze and optimize their decision-making, ensuring that decisions are made in a timely and data-driven manner. This is why when dealing with the enterprise modeling within Industry 4.0, the conceptual model of the decision view can be considered as a core model.

Figure 3.4 illustrates the conceptual model of decision view of an MS formed by the CIAOC. It is worth pointing that each decision can have several constraints, inputs, assumptions, outputs, and criteria. Furthermore, an output is related to only one decision, while many decisions can share the same constraints, inputs, assumptions, and criteria.

Each decision CIAOC property contains a set of information. In the following, we introduce a new concept, that of *MS feature* to structure this information.

Table 3.1: Classification of decisions according to the time horizon, problem family, problem branch, and activity.

ID	Decision problem	Time horizon	Problem family	Problem branch	Activity type
1	Production Strategy	S	Production Strategy	MS Design	Production, Supply
2	Capacity Investment	S	Capacity Planning		Production
3	Aggregate Planning	T			
4	Capacity Allocation	O			
5	Choice of Type of Assembly Line	S	Line Configuration		
6	Line Balancing	S			
7	Equipment Selection	S			
8	Facility Layout	S	Layout Design		
9	Line Layout	T			
10	Line Rebalancing	T	Line Reconfiguration	MS Management	
11	Line Relayout	T			
12	Maintenance Strategy Selection	S	Maintenance and Process Control		
13	Maintenance Load Forecasting	T			
14	Maintenance Capacity Planning	T			
15	Maintenance Scheduling	O			
16	Maintenance Control	O			
17	Sequencing and Scheduling	O	Sequencing and Scheduling		Production, Supply
18	Lot Sizing and Scheduling	T-O	Lot Sizing and Scheduling		
19	Sizing	S	Feeding System Configuration	MS Feeding	Supply
20	Location Planning of Feeding System	S			
21	Routing	T			
22	Scheduling	T			
23	Line feeding Policy	T	Line feeding Policy		
24	Loading	O	Loading		Supply, Storage
25	Inventory Policy	T	Inventory		
26	Inventory Planning and Control	O			
27	Storage Equipment Selection	S	Equipment Selection	Warehouse Design	Storage
28	Material Handling Selection	S			
29	Warehouse dimensioning and sizing	S	Warehouse dimensioning and sizing		
30	Storage Layout	S	Department Layout		
31	Block Staking Problem	S			
32	Configuration of the AS/RS	S			
33	Zoning size problem	T	Zoning	Warehouse Management	
34	Zoning assignment problem	O			
35	Storage Strategy	T	Operation Strategy		
36	Picking Strategy	T			
37	Routing policies	T			
38	Storage Location Assignment Problem	T-O	Storage Location Assignment Problem		
39	Routing & Sequencing	O	Order Picking		
40	Batching	O			
41	Workforce Assignment	O			

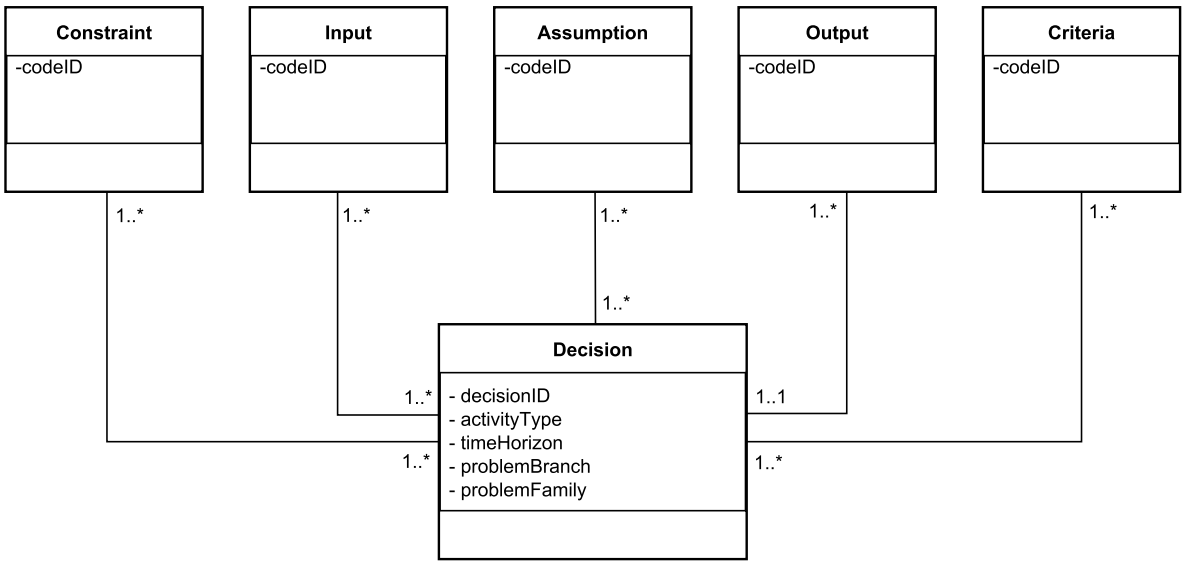

Figure 3.4: The conceptual model of the decision view of an MS.

3.4.2 Conceptual Model of Information View of an MS

Decision properties always belong to what in the following we call *MS features*. MS features can be related to cost (for instance, inventory holding costs), or time (like processing time of tasks), or product/process (for instance, the frequency of product change in a multi-model production line) according to the model choices we made based on Labrousse and Bernard (2008); Neely et al. (1995).

In order to be able to describe CIAOC properties of each decisions in terms the MS features they are made up by, we need to be able to associate each MS feature with a code. For each feature, we define a code composed of two letters and a number. The first letter refers to one of the six classes cited above (P_sP_tRE-CT), and the second refers to the category. The number is used as a unique ID to make the feature identifiable. Subscripts s and t have been added to the letter P to distinguish Process and Product, respectively.

Figure 3.5 illustrates the conceptual model of the information view of an MS object. We could see that we represent each information in MS through *Feature*. The association in Figure 3.5 shows that *Event* occurred in the enterprise impacts on the MS features.

Appendix B provides an exhaustive list of MS features grouped into six classes (P_s, P_t, R, E, C, T) represented in (Tables B.2, B.3, B.4, B.5, B.6 and B.7), respectively. Table B.1 defines abbreviations and code of each class and category that are used in all six tables. Based on these MS features, Subsection 3.4.4 provides the decision problems properties.

The information view links different views in enterprise modeling, as depicted in the following.

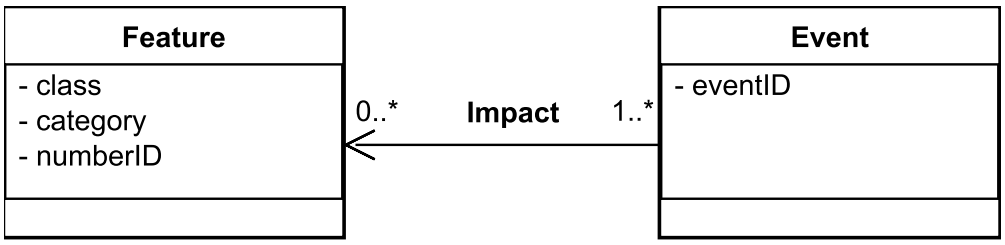

Figure 3.5: The conceptual model of the information view of an MS.

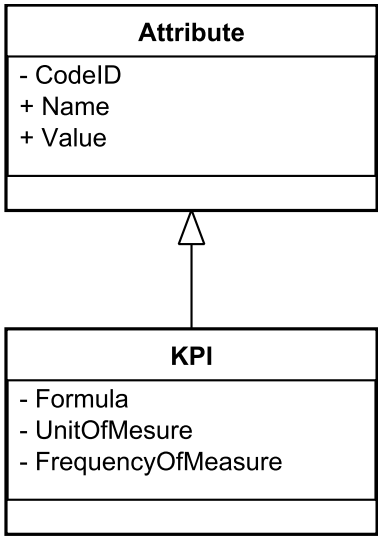

Figure 3.6: The conceptual model of the performance view of an MS.

3.4.3 Conceptual Model of Performance View of an MS

Figure 3.6 illustrates the conceptual model of the performance view of MS. We define two classes Attribute and KPI, linked with inheritance relationship. Each KPI is an attribute having a name, value, formula, unit of measure, and frequency of measure, and an attribute is an element that provides the MS feature value. Thus, we could deduce a relationship between a performance view and information view (c.f. Figure 3.5) as illustrated in Figure 3.7.

Let us consider an example of the productivity of MS to explain the conceptual model of the performance view. Imagine that a company uses ten resources to produce 80 products/day, and the working time is 8 hours/day. Table 3.2 provides an example of instances of attributes created to compute the productivity KPI.

To ensure modeling consistency, we will integrate the already defined views. Figure C.1 illustrates the conceptual model that integrates the four conceptual models.

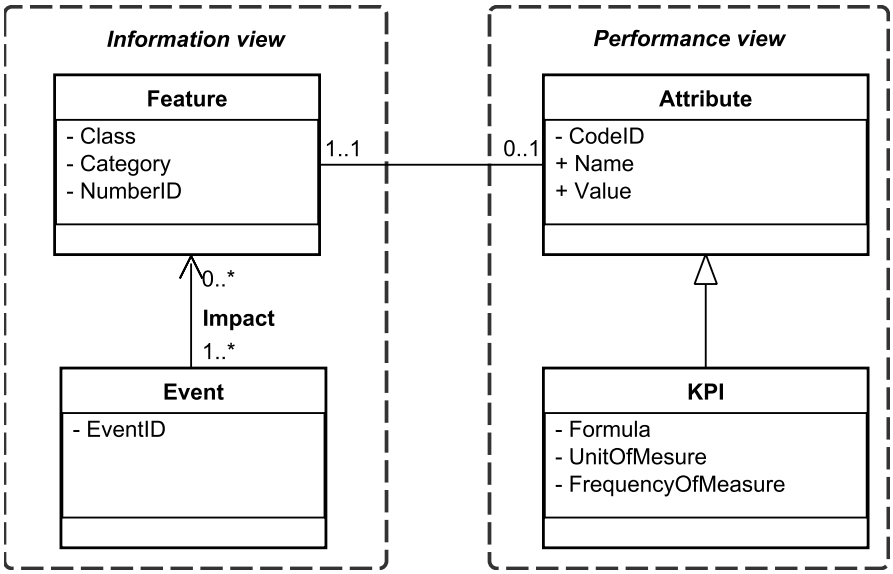

Figure 3.7: The integrated model of the performance and information views of an MS.

Table 3.2: Example of MS productivity.

Instances of attribute			Instance of KPI				
CodeID	Name	Value	Name	Formula	UnitOfMesure	FrequencyOfMeasure	Value
P.p.17	ProducedQuantity	80	LaborProductivity	$\frac{ProducedQuantity}{NumberOfResources \times DailyWorkingTime}$	[#/hour]	Daily	1
R.g.2	NumberOfResources	10					
T.g.4	DailyWorkingTime	8					

3.4.4 Properties of Decision Problems

As already explained, we use a unique code of two letters and one number "*Class.Category.NumberID*" as a notation to refer to each MS feature. A table can be created that uses this notation to specify the CIAOC properties of each problem decision and provide key references. For the sake of space and comprehension, we move this table in appendices with the other tables that define the MS features. So, Table B.8 in Appendix B is dedicated to the definition of CIAOC properties.

Step two defined the CIAOC properties of identified decision problems. Subsequently, the final step will investigate decision interactions.

3.5 Decision Interactions

Decisions can sometimes be considered pairewise and decision interactions can occur: this is the case when we consider two decision problems A and B, whenever A impacts on decision B, or vice-versa, or A must necessarily be solved before B, or they can be addressed jointly.

To see how the knowledge of the CIAOC properties allows to establish the interactions among decision problems, let us consider an example of a decision problem, e.g., the Zoning Assignment Problem (ZAP). Figure 3.8 shows the ZAP and the other decision problems it interacts with. The brown boxes refer to the decision problems apart from the ZAP, in red color, while a link between two boxes highlights an interaction and specifies the interaction type. Arrows indicates *sequential* interactions; dotted-end edges represent *correlated* interactions; finally, double edges denote *simultaneous* interactions.

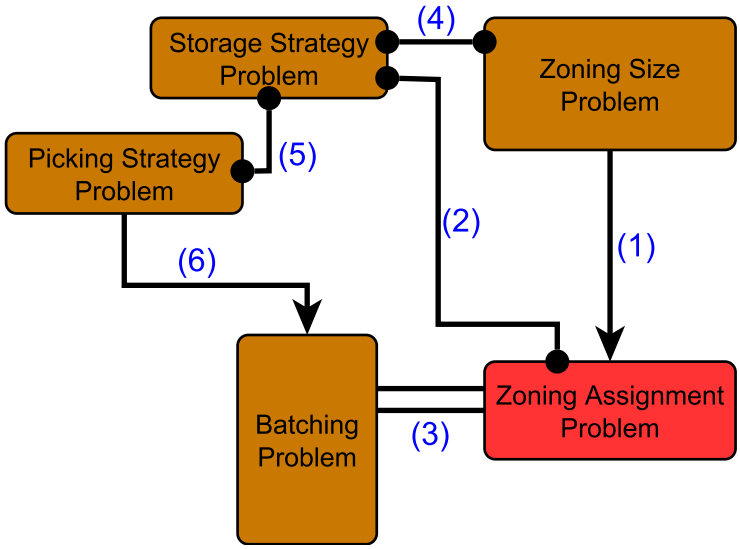

Figure 3.8: Interactions of the zone assignment.
Correlated interaction ⟶ Sequential interaction ═ Simultaneous interaction

In the following, we explain how the CIAOC properties allows to trace such a diagram, to detect and classify decision interactions, and what each type represent.

As stated in Subsection 2.2.3.6, the ZAP consists in deciding the assignment of zones to SKUs and pickers to zones. To solve the ZAP it is necessary to know the number of pickers per zone and the capacity of each zone, which are commonly determined by solwing the zoning size problem (c.f. Subsection 2.2.3.6). To illustrate the interaction of the ZAP with the zoning size problem, Figure 3.8 uses link (1) represented by an arrow, meaning an interaction of *sequential* type.

The interactions explained in the example of the ZAP are illustrated in the Table 3.3, in which we use the already defined code with the format Class.Category.NumberID to refer to each feature of the CIAOC properties.

De Koster et al. (2012) show that interaction exists between the ZAP and the storage strategy problem. More precisely, the assignment of zones to SKUs is among the ZAP decisions impacted by the choice of the storage strategy, i.e., whether we store SKUs using a dedicated or a random location. However, the decision maker can choose among a set of classical storage strategies, therefore solving the ZAP does not require solving the storage strategy problem. As a consequence, we can consider the output of the storage strategy problem as an assumption of the ZAP. To illustrate this, Figure 3.8 uses link (2) represented by a line with a dot at each end, meaning an interaction of type *correlated* exists between the two problems.

When dealing with the ZAP, another assumption to consider is whether batching is applied or not, i.e., whether the arrival orders for picking are grouped and released jointly or not. In other words, when solving the ZAP, one of the outputs of the batching problem is considered an assumption. However, in order to solve the batching problem, we need the assignment of pickers to zones because each picker has a capacity that will impact the batch size and, therefore, the assignment of orders to batches. To illustrate the interaction between the ZAP and the batching problem, Figure 3.8 uses link (3) depicted as two parallel lines, meaning that a *simultaneous* interaction exists between them. Yu and De Koster (2009) is among the works that consider the zone assignement and batching simultaneously.

Links (4), (5), and (6) represent interactions that do not concern the ZAP but rather the problems that are adjacent to it. In the following, we briefly describe them.

When solving the zoning size problem, the decision maker must make some assumption about the storage strategy since he will probably allocate more capacity to each zone in case a dedicated strategy is chosen. Hence, a correlated interaction exists between the decisions about storage strategy and the zoning size, as illustrated by link (4) in the Figure 3.8. Analogously, when deciding the storage strategy, an assumption about the picking strategy is required as it impacts the number of pickers per zone. Hence, the interaction between the storage and picking strategies can be classified as being of

type correlated, as shown by link (5). Finally, the sequential interaction between picking strategy and batching problem (link (6)) can be explained by considering that using a specific picking strategy such as the single order picking will impacts the decision of grouping orders, i.e., the batching. More precisely, the batch size will depend on the used picking strategy, e.g., whether a concurrent picking strategy is used. That is why the picking strategy is previously solved and is considered to be a constraint of the batching problem.

In general, and unlike constraints, inputs, assumptions, and outputs, criteria do not enter directly into the determination of the interactions. However, they can be used to identify decisions that have some criteria in common to optimize. For example, ZAP and batching, have a criterion in common consisting in minimizing the balance of the workload (R.o.4), which reinforces the idea of integrating these decisions. Of course, a simultaneous interaction may exist between two decisions despite not having a common criterion. However, most often, there would be a relationship between their criterion.

Based on CIAOC properties, Table 3.3 provides also insights about the decision nature. These properties are conveniently grouped into categories: Cost, Time, Product, Process, Resource, and External effect. By analyzing these properties, we can identify the data relevant to a specific decision problem and infer the impacted categories when making the decision.

3.5.1 Some Decision Interactions Linked to Plant Activities

In what follows, we present some examples of how decisions related to each activity (production, supply, and storage) interact with each other.

3.5.1.1 Some Interactions of Production Activity Decisions

For instance, the production strategy decision is made before the inventory policy and the capacity investment decisions. As specified in Subsection 2.2.2.3there are two inventory policies, the continuous-review and the periodic-review. For continuous-review models, most studies assume the First Come First Serve (FCFS) allocation rule, and the probability density function varies according to the lead time demand, as mentioned by Rizkya et al. (2018). In contrast, for periodic-review models, since the demand is accumulated and fulfilled at the end of each period, different allocation rules are considered, such as fixed priority, FCFS, and fair-share rules. In both cases (periodic and continuous), the total inventory cost is frequently the criteria for optimization. Our review indicates that the inventory policy models suitable for Assembly-To-Order (ATO) may not be appropriate for Make-to-Stock (MTS) or hybrid strategies. For instance, Benjaafar and ElHafsi (2006) demonstrate that in ATO systems, determining the optimal base stock level and rationing level of each component relies on the inventory levels of other components. However, in MTS, choosing the periodic review

Table 3.3: Interactions of zoning assignment problem illustrated by the CIAOC properties of decision problems.

Problem	Constraints	Inputs	Assumptions	Outputs	Criteria
Zoning size	Picking capacity of picker (R.c.8) Warehouse size (R.c.23) Storage layout (R.c.24) ...	Number of aisles (P.β.9) Set of orders (P.Π.4) Set of all items (Z.Π.5) Set of pickers (R.Π.4) ... ... (1)	progressive or parallel zoning (P.a.35) All zones are of the same size (R.a.42) Dedicated, Random Storage (P.w.13) Travel time between zones is negligible (T.a.25) ...	Number of zones (R.g.8) Length and width of zones (R.w.23) Number of pickers per zone (R.c.17) Capacity of each zone (R.c.25)	Minimize the system throughput time (T.o.14)
Zoning assignment	Number of pickers/zone (R.c.17) Warehouse size (R.c.23) Storage layout (R.c.24) Capacity of each zone (R.c.25)	Set of zones (R.Π.3) Set of SKUs (Z.Π.6) Set of orders (P.Π.4) Set of pickers (R.Π.4) ...	progressive or parallel zoning (P.a.35) Similar pickers skills (R.a.49) Batching or not (P.a.29) Dedicated, Random Storage (P.w.13) ... (2)	Assignment of zones to SKUs (R.w.24) Assignment of pickers to zones (R.w.25) (4)	Minimize the workload balance (R.o.4) Maximize the crane throughput (R.o.11) Minimize the number of zones visited (R.o.14)
Storage strategy	Total storage capacity (R.c.19) Storage layout (R.c.24) Sevrice requirements (E.c.5)	Skewness of demand (E.a.8) Input and output flows (P.w.1) Periodic,continuous review policy (P.s.5)	Unit load Warehouse or AS/RS (R.a.55) Single, dual or multi-command (P.a.24) Single, Sequential, Concurrent picking (P.w.14) (5)	Dedicated, Random Storage (P.w.13) (3)	Minimize travel time (T.o.16) Minimize the traveling costs (C.o.19)
Picking strategy	Storage layout (R.c.24) Zone configuration (P.β.12) Storage assignment (Z.c.6)	Characteristics of product (Z.c.1) Number of orders per day (P.θ.1) Warehouse configuration (R.w.10) (6)	Batching or not (P.a.29) Routing and wave length are used (P.a.44)	Single, Sequential, Concurrent picking (P.w.14)	Maximize the storage efficiency (P.o.7)
Batching	Time slots (T.c.9) Order due dates (T.c.10) Picking capacity of picker (R.c.8) Single, Sequential, Concurrent picking (P.w.14)	Picker's travel speed in zones (R.w.22) Size of pick bins (R.w.8) Set of orders (P.Π.4) Set of location of each order (P.Π.5) ...	Static or dynamic batching (P.a.30) Level of partitionning (P.a.34) The orders are picked one at a time (P.a.36)	Size of each batch (P.w.16) Assignment of orders to baches (P.w.17) Batch time window (T.w.10)	Minimize travel time (T.o.16) Minimize the picker effort (R.o.12) Minimize the workload balance (R.o.4)

policy with a fixed order quantity is more adapted since there is less variability. Therefore, the production strategy impacts on selecting an inventory policy and thus choice of the appropriate model.

Likewise, the capacity investment decision must be made before the aggregate planning decision. According to Rafiei et al. (2013) the production strategy is the first level of the hierarchichal production planning, then in the second level the capacity planning are made, and finally the scheduling and sequencing decisions. The production strategy could be seen as an assumption to solve the capacity planning decision. For example, Mincsovics et al. (2009) consider the MTS strategy and develop a model to evaluate the balance between the capacity and inventory economically, and Wang and Nguyen (2017) assume MTO strategy and study the investment cost of resources of new technology. All these reasons, show that the production strategy decision is made before the capacity planning. Furtheremore, depending on single or multi-product and single or mutlti-periods the models to use to choose the optimal production are different (Bollapragada et al., 2004). Xiao et al. (2010) investigate under ATO environment with both uncertain demand and uncertain assembly capacity, the impact on the components quantity to stock and the quantity to assemble-in-advance. That means that the aggregate planning decision considers the production environement with both demand forecasting and capacity constraint. In other words, they are previous decisions of the aggregate planning decisions.

As described before, the line layout assigns the machines to the locations available in each period on the shop floor (Meng et al., 2004). Since the ALBP is a strategic decision that allows obtaining the number optimal of machines, it is common to solve the ALBP before the LLP.

In the recent literature, the lot sizing decision is not always considered alone but instead is jointly considered with the scheduling decision. For example, lot sizing and scheduling are often considered simultaneously in a comprehensive way (Schemeleva et al., 2018, 2012). The goal is to determine the starting and completion time of each product in each period on each machine and the quantity produced by the shop in each period. For more details, the reader can refer to Maravelias and Sung (2009).

3.5.1.2 Some Interactions among Supply Activity Decisions

Battini et al. (2010) deal with the LPFS, and determine the number of supermarkets and Line feeding policy decisions at the same time. Emde and Schneider (2018) address the sizing and routing problems jointly. They use a mathematical formulation in order to minimize the number of vehicles and the total duration of all routes.

Satoglu and Sahin (2013) solve the scheduling in the supply activity and the routing decision problems jointly assuming only one central warehouse feeding the workstations.

They formulate the problem as a non-linear mixed-integer program and use an heuristic to minimize the inventory cost and the number of handled parts.

Fathi et al. (2015) propose a mathematical model to minimize the number of tours and inventory level of components at the border of the line under a vehicle capacity constraint. Using a modified particle swarm optimization algorithm, they solve the scheduling and loading problems at the same time.

3.5.1.3 Some Interactions among Storage Activity Decisions

The storage strategy is considered as an assumption to solve the AS/RS configuration in several articles. For instance, Bozer and White (1984) and Chang and Wen (1997) cope with the configuration of AS/RS under the assumption of using a randomized storage strategy. Note that the configuration of AS/RS includes the specification of the S/R machine travel time. According to Park et al. (1999), if no assumption about this specification is made, their results obtained about the throughput of AS/RS remain valid under various storage strategies. Hausman et al. (1976) provide the expressions to compute the travel time for random storage and turnover-based storage strategies.

Won and Olafsson (2005) formulate the batching and picker routing problem jointly as a combinatorial optimization problem. Likewise, Kulak et al. (2011) focus on an integrated approach for order batching, routing and sequencing. They propose a cluster-based algorithm and a tabou search algorithm.

Goh et al. (2001) assume a dedicated storage policy to solve the warehouse sizing. They state that in the case where the sizing and dimensioning decisions depend only on the storage cost, the randomized storage policy can be more appropriate. This highlights a possible link between the dimensionning and sizing decision and the storage policy decision.

Larson et al. (1997) propose an heuristic to compute the length and depth for each storage zone taking into account the configuration of the input/output points. They assume that the zone length is identical for all storage zones. Hence, it can be beneficial to take the storage layout decision prior the zoning size decision.

We resorted to both the literature and rules to study decision interactions. The literature makes it possible to analyze how decisions were taken into account in the past and identify the types of existing interactions. In addition, it serves as a guide to define rules capable of identifying the interaction type based solely on the CIAOC properties of decision.

3.5.2 Types of Decision Interactions

Based on the literature review, we define three types of interactions among decisions as briefly explained in Subsection 3.2.1. Below we provide more detail about each interaction

type.

- correlated interaction: it occurs between two decisions that affect each other without there being a temporal sequence or dependence between them. In the literature, authors do not agree on the sequential order of making these decisions.
- sequential interaction: it occurs between two decisions for which an order needs to be respected. For interactions of this type, usually, most authors in the literature agree.
- simultaneous interaction: this is the case of decisions that can be taken at the same time and are often considered integrated into the literature. We will also consider as simultaneous decisions, s.t. even if no work in the literature integrates actually them, many works exist proving the advantages of their integration.

The MS type (see Subsection 1.2.1.1) is another significant factor impacting the type of decisions interactions. We add two additional interaction types, DMS-only sequential and FMS/RMS-only sequential interactions. The goal is to cover some cases where a distinction between DMS and FMS/RMS is required. In Subsection 3.5.4, we give examples of such cases.

Based on the literature, and to the best of our knowledge, Table 3.4 provides the existing interactions among all decision problems related to the MS. Table 3.4 uses symbols defined for the intearction types to illustrate the interaction among two decision problems (depicted as Problem 1 and Problem 2). The column references specifies the works from the literature discussing this interaction type. The black dotted arrow and the red dotted arrow are conventions to distinguish the two types of additional interactions, respectively, DMS-only sequential and FMS/RMS-only sequential interactions, which we will explain in the following.

For example, the types of interactions depicted in Figure 3.8, involving the Zoning Assignment Problem (ZAP) and other decision problems, are also presented in Table 3.4. Additionally, references are provided for each interaction between two decision problems illustrated in Figure 3.8. For instance, Melacini et al. (2010) demonstrate the link(1) between the zoning size problem and the ZAP. The authors focus on developing a framework for designing pick-and-pass order picking systems. They consider the number of zones and the number of pickers as design variables. Subsequently, they address the assignment of items to the zones.

3.5.3 Rules to Study Decision Interactions

In the following, we define a set of rules that will help us identify decision interactions and determine their type. Through the analysis of decision interactions in the literature, our study reveals that the governing rules can be categorized into two distinct types:

Table 3.4:
Interactions of decision problems based on the literature.
●—● Correlated interaction ⟶ Sequential interaction ═══ Simultaneous interaction
– – ➢ DMS-only sequential interaction – – ➢ FMS/RMS-only sequential interaction.

Problem 1		Problem 2	References	Problem 1		Problem 2	References
Production strategy	⟶	Inventory policy	(Benjaafar and ElHafsi, 2006)	Routing	═══	Scheduling	(Satoglu and Sahin, 2013)
Production strategy	⟶	Capacity investment	(Rafiei et al., 2013)	Scheduling	═══	Loading	(Fathi et al., 2015)
Capacity investment	⟶	Aggregate planning	(Bollapragada et al., 2004)	Loading	⟶	Inventory planning & control	(Emde et al., 2012)
Capacity investment	– – ➢	Line balancing	(Deif and ElMaraghy, 2006)	Line feeding policy	⟶	Inventory policy	(Choi and Lee, 2002)
Aggregate planning	⟶	Capacity allocation	(Vercellis, 1991)	Inventory policy	⟶	Storage strategy	(Ang et al., 2012)
Aggregate planning	⟶	Lot sizing	(Weinstein and Chung, 1999)	Inventory policy	⟶	Inventory planning & control	(Singha et al., 2017)
Aggregate planning	– – ➢	Line rebalacing	(Deif and ElMaraghy, 2006)	Inventory planning & control	⟶	Scheduling	(Laguna, 1999)
Capacity allocation	⟶	Scheduling	(Akcali and Uzsoy, 2000)	Dimensionning-Sizing	⟶	Storage Layout problem	(Roodbergen and Vis, 2006)
Facility layout	⟶	Choice type of line	(Ullah et al., 2014)	Dimensionning-Sizing	●—●	Storage strategy	(Gu et al., 2010)
Facility layout	⟶	Line layout	(Drira et al., 2007)	Storage layout	⟶	Zoning Size	(Larson et al., 1997)
Facility layout	⟶	Sizing & Dimensioning	(Heragu et al., 2005)	Storage layout	⟶	Storage location assignment	(Rojas-Reyes et al., 2019)
Choice type of line	⟶	Line balacing	(Thomopoulos, 2014)	Storage layout	●—●	Storage strategy	(Bassan et al., 1980)
Line Layout	⟶	Line relayout	(Kochhar and Heragu, 1999)	Storage layout	⟶	Block stacking	(Derhami et al., 2016)
Line Layout	⟶	Location planning of feeding system	(Alnahhal and Noche, 2015b)	Storage layout	⟶	Configuration of AS/RS	(Roodbergen and Vis, 2009)
Line balancing	⟶	Line layout	(Meng et al., 2004)	Storage equipment selection	●—●	Material handling selection	(Mohsen, 2010)
Line Balancing	═══	Sequencing & scheduling	(Sawik, 2002)	Storage equipment selection	⟶	Block stacking	(Gu et al., 2010)
Line Balancing	– – ➢	Scheduling	(Sawik, 2002)	Storage equipment selection	⟶	Configuration of AS/RS	(Roodbergen and Vis, 2009)
Line balancing	═══	Equipement selection	(Özdemir and Ayağ, 2010)	Configuration of AS/RS	●—●	Storage strategy	(Hausman et al., 1976)
Line balancing	⟶	Line rebalancing	(Makssoud et al., 2015)	Block stacking	⟶	Storage location assignment	(Yang and Kim, 2006)
Line rebalancing	⟶	Line relayout	(Zha and Yu, 2014)	Storage location assignment	⟶	Batching	(Ene and Öztürk, 2011)
Line rebalancing	– – ➢	Sequencing	(Grangeon et al., 2011)	Material handling selection	⟶	Sizing	(Chang et al., 2013)
Lot-sizing	═══	Scheduling	(Urrutia et al., 2014)	Material handling selection	⟶	Line feeding Policy	(Baller et al., 2020)
Sequencing	═══	Scheduling	(Prabaharan et al., 2006)	Zoning size	⟶	Storage location assignment	(Gray et al., 1992)
Maintenance strategy	═══	Line balancing	(Meng et al., 2021)	Zoning size	⟶	Zoning assignement	(Melacini et al., 2010)
Maintenance strategy	⟶	Line rebalacing	(Meng et al., 2021)	Zoning assignment	═══	Batching	(Yu and De Koster, 2009)
Maintenance strategy	⟶	Maintenance forecasting	(Grall et al., 2002)	Storage strategy	●—●	Zoning size	(Le-Duc and De Koster, 2005)
Mainetanance load forecasting	⟶	Maintenance capacity planning	(Al-Shayea, 2012)	Storage strategy	⟶	Routing & sequencing	(Boysen et al., 2017)
Maintenance capacity planning	⟶	Maintenance scheduling	(Duffuaa and Raouf, 2015)	Storage strategy	●—●	Zoning assignement	(De Koster et al., 2012)
Maintenance scheduling	═══	lot sizing	(Lee and Rosenblatt, 1987)	Storage strategy	●—●	Picking strategy	(Petersen II, 2009)
Maintenance scheduling	═══	Sequencing	(Sortrakul et al., 2005)	Storage strategy	●—●	Routing strategy	(Petersen, 1997)
Maintenance scheduling	⟶	Maintenance control	(Ben-Daya et al., 2016)	Routing strategy	⟶	Routing & sequencing	(Chen et al., 2015)
Inventory planning & control	═══	Maintenance scheduling	(Lee, 2005)	Routing strategy	●—●	Picking strategy	(Glock and Grosse, 2012)
Location planning of feeding system	═══	Line feeding policy	(Schmid and Limère, 2019)	Picking strategy	⟶	Batching	(De Koster et al., 2007)
Location planning of feeding system	⟶	Sizing	(Faccio et al., 2013)	Batching	═══	Routing & sequencing	(Chen et al., 2015)
Sizing	═══	Routing	(Emde and Schneider, 2018)	Routing & sequencing	⟶	Workfoce assignement	(Rubrico et al., 2011)
Routing	⟶	Line feeding policy	(Baller et al., 2020)	Workfoce assignement	⟶	Loading	(Rubrico et al., 2011)

- generic rules: established based on unanimous conclusions and a consensus found within the existing literature. They serve as a foundation for understanding decision interactions and are widely acknowledged in the field.

- specific rules: derived from our comprehensive investigation and are formulated based on specific criteria unique to our research. They provide additional insights and considerations beyond the generic rules, contributing to a more nuanced understanding of decision interactions.

3.5.3.1 Generic Rules to study decision interactions

Two decision problems A and B are said to be in correlated interaction if:

- one of A's outputs is one of B's assumptions.

- one of A's outputs is one of B's constraints, and one of B's outputs is an assumption of A.

Two decision problems A and B are said to be in sequential interaction if:

- one of A's outputs is also one of B's inputs.

- Using the above rules, we find that the interaction between A and B is correlated and, at the same time, is sequential.

Two decision problems A and B are said to be in simultaneous interaction if:

- Significant works exist in the litterature considering the integration of decision problems A and B.

3.5.3.2 Specific Rules to Study Decision Interactions

Two decision problems A and B are said to be in sequential interaction if:

- Using the generic rules, we find that the interaction between A and B is sequential in both orders. Then, we will consider the interaction between A and B sequential in one-way. We resort to the literature to find the sequence respected by most authors.

Two decision problems A and B are said to be in correlated or sequential interaction if:

- one of A's outputs is one of B's inputs or constraints, and one of B's outputs is one of A's inputs or constraints. Then, the interaction between A and B is either correlated or sequential in one-way (based on the literature).

Two decision problems A and B are said to be in simultaneous or sequential interaction if:

- one of A's outputs is one of B's constraints and none of B's outputs are part of A's assumptions.
- one of A's outputs is one of B's inputs, and one of B's outputs is an assumption of A.

Others rules including either the transitivity or the priority rule:

- one of A's outputs is one of B's inputs or constraints, and one of B's outputs is one of C's inputs or constraints. Suppose one of the features of A is inputs or constraints of C. Then, we consider only the interaction between A and B and the interaction between B and C. Since we need to make decision B before decision C. In other words, if decisions (A - B) and (B - C) have the same interactions and are either sequential or simultaneous, then we assume that the transitivity explains any interaction between A and C.
- Simultaneous interaction is stronger than sequential interaction, and sequential is stronger than correlated. The term stronger indicates that if works exist that consider A and B simultaneously and other works consider them sequentially, we will consider that the interaction between A and B is simultaneous. We deduce the same conclusion about sequential and correlated interactions.

3.5.4 MS Type-Dependent Decision Interactions

As already mentioned, the MS type may impact the interaction among decisions. We have identified two MS-dependent decision interactions, namely DMS-only sequential and FMS/RMS-only sequential: The first specifies that the interaction between two decisions is sequential but only within the context of the DMS type, while, the second refers to sequential interaction but specifically within the FMS/RMS type.

Below, we present decisions concerned by these interaction types to figure out how the MS type may influence the interaction between two decisions.

3.5.4.1 Interaction between the Line Balancing and the Sequencing & Scheduling

On the one hand, Emde et al. (2009) propose three alternatives to consider the line balancing and model sequencing problem, namely successive planning, simultaneous planning, and anticipation. The successive planning shows a case of sequential interaction, where the line balancing is solved first, then the model sequencing is decided. The anticipation planning is also a case of sequential interaction, but when solving the line balancing, we consider the coordination between the two decisions, and the impacts on the sequencing are taken into account in an aggregated manner because the time horizon of the two problems is not the same. Lastly, the simultaneous planning

considers a simultaneous interaction, and for solving this case, Boysen et al. (2009) subdivide the approaches used in the literature into three techniques (mixed-model sequencing, car sequencing, and level scheduling).

On the other hand, Sawik (2002) distinguish two approaches to address the two problems, monolithic and hierarchical, corresponding to what in our study we have named simultaneous and sequential interactions, respectively. The authors conclude that using the MIP formulations in low volume and low mix production provides good results for both a monolithic and hierarchical approach. However, for the large volume, which is the case of DMS, a greater computational effort is required, and a hierarchical approach is usually preferable.

Simultaneous interactions may be possible when the time horizon of two decisions belongs to the same time horizon or two successive time horizons. Since the time horizon of the line balancing, sequencing and scheduling does not respect this condition. It means that considering the two decisions simultaneously is not possible. However, there is a work that successfully integrates them in the context of FMS by considering both decisions as short-term problems. We decide to consider the two decisions as simultaneous for two reasons: First, this section deals with interaction type-dependent decision interactions, and we want to distinguish the case of DMS regarding FMS/RMS. Second, as we have shown in X dedicated to the evolution of the time horizon of decisions, the line balancing problem will switch from the strategical level to the strategical-tactical level, thus, it makes sense to keep this interaction.

Hence, based on this statement and using the generic rules, we determine that the interaction between line balancing and sequencing (including scheduling) is simultaneous, as at least one work integrates the two decisions. This finding holds for all MS types, including DMSs, FMSs, and RMSs. However, it is essential to emphasize that for DMSs specifically, the sequential interaction between line balancing and scheduling proves to be more advantageous. Consequently, in this case, we classify the interaction as DMS-only sequential. Furthermore, we explicitly state that the sequencing problem is not considered when examining DMSs.

3.5.4.2 Interaction between the Capacity Planning and the Line Balancing/Rebalancing

As noted by Deif and ElMaraghy (2006), we distinguish three approaches to address capacity planning problems.

The first approach involves fixing capacity based only on forecast demand, as it is the case of DMSs and mass production. The second approach is called capacity on-demand, i.e., adapting the capacity at each period due to the reconfigurability characteristic of the MS, in the case of FMS/RMS. Finally, in RMSs, we adopt the adaptive capacity approach that combines the two previous ones.

Line balancing is based on the capacity investment decision and, based on the conclusions of the mentioned work, DMS-only sequential interaction exists between the capacity investment and line balancing. In the case of FMS/RMS, however, the aggregate planning could be changed at each period, and then the line rebalancing can be performed. Hence, an FMS/RMS-only sequential interaction exists between the aggregate planning and the line rebalancing.

3.6 A Graphical Tool to Represent Decisions in MSs and How they Interact with Each Other

All interactions between existing decisions have been identified based on the literature and possibly further specified based on the rules of Subsection 3.5.3. Although Table B.8 is helpful to make an inventory of interactions, a cartography of decisions better illustrates the taxonomy of decision problems and their interactions. The results are shown in Figure 3.9. This cartography of decisions is a results synthesis obtained by the three-step procedure. In the following, we define the graphic conventions that we use in it.

We represent the cartography of decisions as a two-dimensional plan, in which, the vertical axis represents the time horizon, and the horizontal axis, the type of activity as we illustrated in Figure 3.1. As regards the time horizon, we represent going from the strategic level to the operational level from top to bottom. Production, supply, and storage activities are shown from left to right: we choose this convention according to the flow of material.

Yellow boxes represent the problem families, while brown and blue dotted, labeled boxes represent decision problems and problem branches, respectively. Arrows and lines highlight the interactions between decision problems, as well as their type, according to Table 3.4.

All these conventions remain valid when used between problem families.

When a problem family includes problems belonging to different time horizons, the coresponding yellow box overlaps the frontier(s) between the time horizon levels. Similarly, when it includes problems related to two activities, the corresponding yellow box spans over the two corresponding activity sections.

A similar reasoning applies to the decision problems (brown boxes).

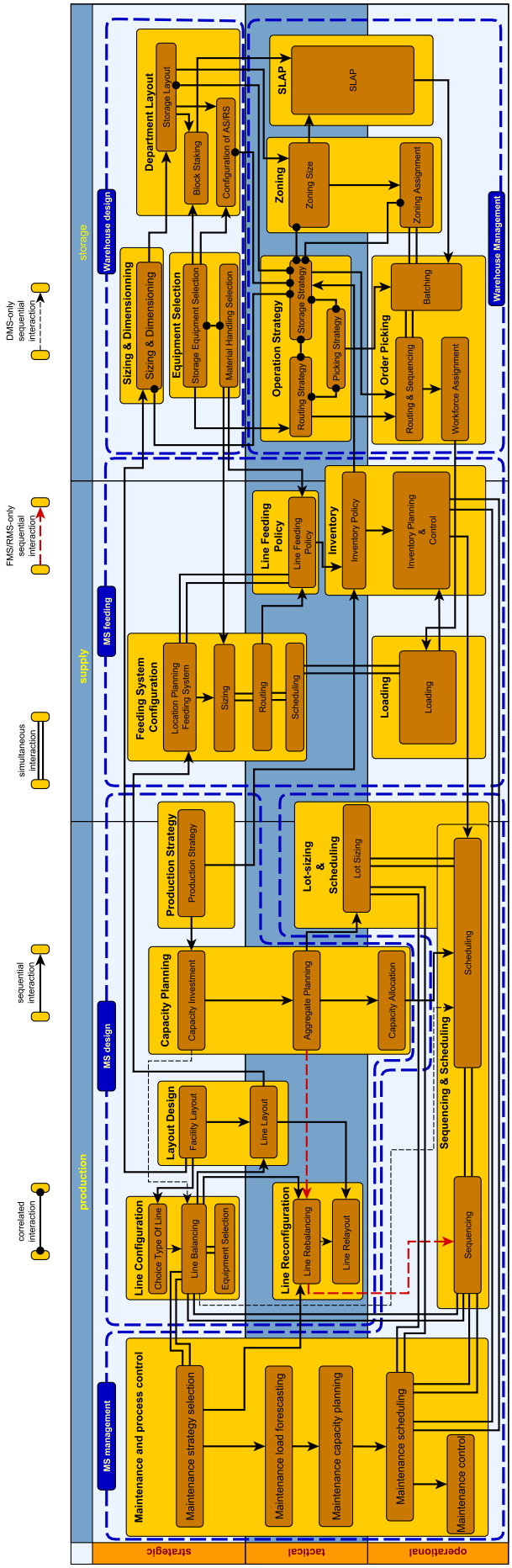

Figure 3.9: Cartography of decisions according to time horizon and activity.

3.7 Using the Cartography Tool to Represent the Evolution of Decision Problems and Interactions among them in the Industry 4.0 Context

In this section, we propose to use the cartography tool presented in Section 3.6 to illustrate decision evolutions in Industry 4.0, in the continuation of what has been presented in Section 2.4. We identify the main evolutions as regards the time horizon and the interactions among decisions; for the MS type, we will assume to be dealing with one that fits the smart factory requirements, such as adaptable MSs (e.g., RMSs with the adaptability characteristic). Hence, the focus will be on time horizon and interactions.

3.7.1 Evolution in the Time Horizon of Decisions

In the following, we will discuss the impact of each other transformation drivers discussed in Section 2.3 and shown in Table 2.1 on the time horizon of decisions. For each transformation driver, a suitable figure similar to the cartography of Figure 3.9 will depict the discussed changes.

Standardization of data exchange

This transformation driver can help accelerating the interoperability among different business systems within Industry 4.0. Smart machines are expected to provide data about their status to Computerized Maintenance Management System (CMMS) in a short-term period. Furthermore, production data from MES allow for tracking daily machine utilization. Based on these data, an analysis of deterioration speed could be conducted in order to forecast maintenance load. We expect the integration of CMMS with MES to be able to forecast the maintenance load on a weekly or monthly basis.

On the other hand, based on the forecasting, the CMMS will be able to exploit data provided by Labor Management System (LMS) concerning the availability of workers and their skills, to propose maintenance schedules for performing maintenance jobs. In classical configurations, it is common to have the Sales & Operation planning (S&OP) as a prerequisite to aggregate planning. Also, the aggregate planning horizon in most industries varies between three months and one year. However, a dynamic environment with high uncertainties involves reducing the planning horizon leading to coping with S&OP instead at the tactical level and aggregate planning at the operational level. This change becomes possible thanks to the interoperability among at least three business systems: Customer Relationship Management (CRM), ERP, and MES.

Even the line layout problem is impacted by the standardization of data exchange. Thanks to data exchange standardization, Machine-to-Machine (M2M) collaboration has become possible. As a consequence, a higher agility can be achieved by adapting the line layout to the production planning. Therefore, it is expected that machines will be

able to autonomously decide its next location within a plant. For instance, according to the production planning, we could switch from flow shop to job shop configuration on a weekly or even daily basis if the plant structure allows one-click deployment.

The standardization of data exchange will also impact the inventory policy and promotes the use of continuous review. Often, the solution of a WMS integrates an IMS module. Our analysis of the Industry 4.0 environment expects the emergence of standalone IMS solutions, but interoperable with WMS. Considering IMS as a standalone solution provides high flexibility. First of all, an IMS aims to track in real-time the inventory of each component everywhere within the enterprise, i.e., at the logistic platform, at the warehouse, at the supermarkets, and at the border of lines. Most solutions use the periodic review as an inventory policy, due to a lack of realistic solutions that would allow a continuous tracking. With more recent IMS solutions, inventory policies tend to be continuous based on smarter rules that use data provided by MES, ERP, and WMS. Therefore, the decision on the inventory policy will shift at the operational level.

Likewise, the standardization of data exchange among the Computer-aided design (CAD), Product lifecycle Management (PLM), and WMS plays a central role in making the storage layout more dynamic, then modified more frequently. In the past, when designing the plant warehouse, the storage layout was a strategic decision involving high investment. With shorter product life cycle (PLC), industrial companies understand the necessity to switch the storage layout decision to medium-term, at the tactical level. The aim is to adapt the storage layout to the current product and the quantity planned to be produced during each period based on customer demand forecasting. It is important to consider the storage layout when designing products in a CAD software. This product design enables the PLM software to generate the engineering bill of materials (EBOM), which includes component specifications like dimensions and weight. The PLM then shares this data with the WMS to initiate the design of the necessary storage layout

The standardization of data exchange among business systems impacts also the storage strategy problem. We expect that the integration of business systems, the Transport Management System (TMS), WMS, and ERP, will allow the adoption of the storage strategy on a short-term basis. This is because the TMS provides the vehicle arrival date and time of the component transported, while, the ERP provides the planned manufacturing order to derive the weekly production planning. Using these data, the WMS can thus decide whether to adopt a dedicated or random storage strategy. That is why the standardization of data exchange will impact the time horizon of storage strategy and lead to considering it at the operational level.

In the same context, the (Storage Location Assignment Problem) SLAP decision deals with assigning a component to the storage locations. We already recalled that two versions of this problem (static and dynamic) are cited in the literature. The static

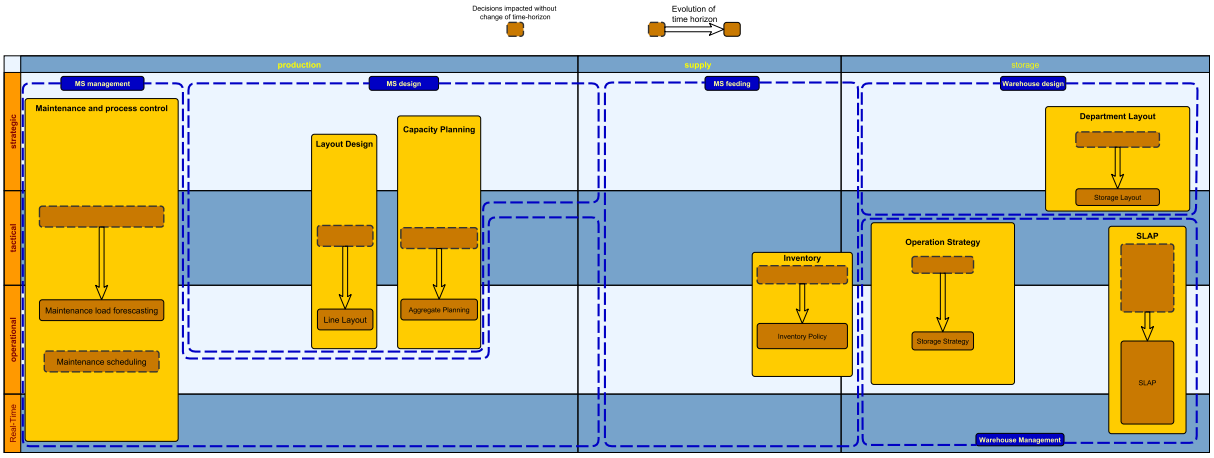

Figure 3.10: The impacts of standardization of data exchange on time horizon of decisions.

version is tackled at the tactical level, while the dynamic version is considered at the operational level. The standardization of data exchange between the WMS and IMS offers the capability to track the availability of storage locations in real-time and to know precisely the inventory level of each component and its location. Therefore, the assignment of locations could use rules to promote the grouping of similar components in the same zone or of components depending on their expected time of consumption.

Figure 3.10 illustrates these discussed changes on decisions resulting from the impacts of the driver: standardization of the data exchange.

New techniques and algorithms

Today, new techniques and algorithms, especially Machine Learning (ML) and Big Data Analytics (BDA), are being widely adopted. These methods involve using novel ML algorithms and BDA, encompassing new modeling approaches. However, despite their potential benefits, the challenge of processing huge amounts of data remains a significant concern. Due to their synergy, this challenge is closely linked to the "Integration of data" challenge.

Research and industry efforts have predominantly focused on developing domain-specific ML models rather than cross-domain ones. Consequently, reusing these models in different contexts is not straightforward. Nevertheless, ML remains compatible with the Adaptable MS concept because it can autonomously learn from the manufacturing environment and adapt to changes that occur.

Implementing new technologies has enabled the collection of huge amounts of data, which requires processing to extract valuable insights. Two main approaches to dealing with big data are batch and streaming. Batch processing involves storing and analyzing data, which is suitable when time is not critical for decision making. On the other hand, streaming processing involves keeping only a portion of the data arriving in a

continuous stream, making it more suitable for real-time decision-making. MapReduce is recognized for its scalability in handling Big Data queries but does not provide real-time capabilities, making it more convenient for batch processing. On the other hand, Kafka is a streaming platform for streaming processing, but it is a more cloud-based solution than an edge-based solution. The preferred solution for real-time decision making in MSs is edge-based. For instance, Shiue et al. (2018) propose a new reinforcement learning-based model to deal with a real-time scheduing.

The impact of these new techniques and algorithms on decision-making in MSs concerns the structure of decisions more than their time horizon. The availability of massive data relaxes assumptions for decision making and introduces new criteria for consideration. For example, in production activities, the maintenance control may shift from an operational horizon to a real-time horizon by using data to predict machine behavior in real time. Maintenance strategy selection may also involve trade-offs between analytics accuracy and reporting time. Based on such considerations, Big Data Analytics can provide recommendations for the most suitable maintenance strategy.

For supply activities, new techniques and algorithms may impact scheduling problems and require addressing them at the operational level, especially if data sharing in real-time is facilitated by Autonomous Guided Vehicles (AGVs) or Autonomous Mobile Robots (AMRs). The line feeding and inventory policy decisions traditionally relied on experience. Still, they can now be informed by historical and event-based spatiotemporal data models to track and trace component stock levels and locations.

In storage activities, decision problems related to storage, routing, and picking strategies, which previously relied on rules, may shift to the operational level with the adoption of AI-based algorithms that outperform traditional heuristics and offer optimal short-term strategy. Moreover, clustering algorithms can be used to derive common patterns from data communicated by smart products or components, thereby impacting storage location assignment and batching decisions. In Figure 3.11, we provide an illustration of how new techniques and algorithms can influence decision problems in manufacturing systems.

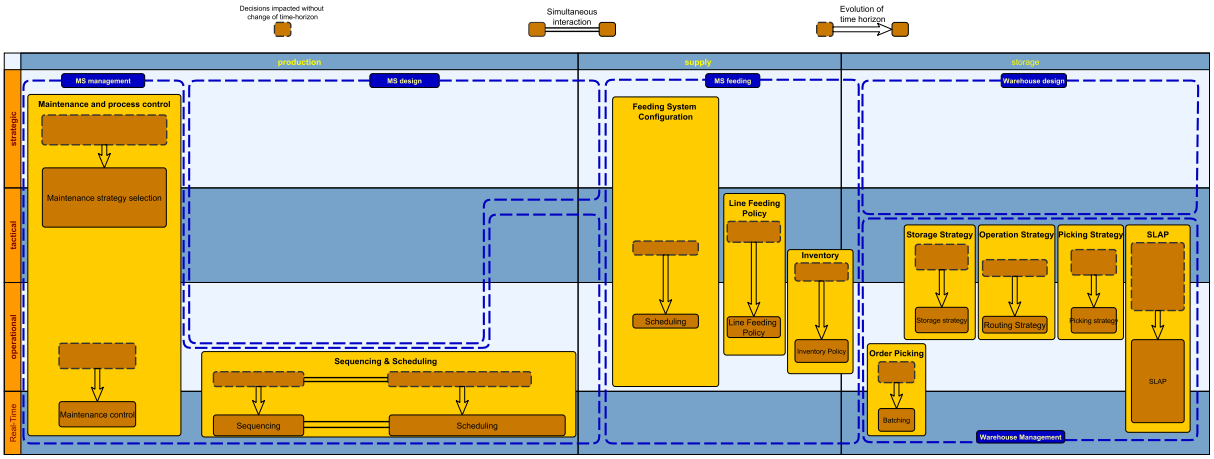

Figure 3.11: The impacts of new techniques and algorithms on time horizon of decisions.

Leagile strategy

The main impact of this transformation driver is on strategic decisions. Even if the decision problems belonging to the family are not impacted, thus not represented, we choose to keep the yellow family boxes with the same form only for homogeneity with the previous presentation done in Figure 3.9.

Among the decision problems whose time horizon will shift, there is the maintenance strategy selection, in the MS management branch. Switching the maintenance strategy may depend on the health and age of equipment, data collected during a specific period, and the occurence of failures. As a result, the time horizon can shift to tactical, except for some critical equipment, e.g., engine of aircraft, for which the maintenance strategy remains strategic as it is decided when purchasing the equipment according to the standard requirements.

Also, the leagile strategy leads to dealing with the production strategy problem differently. Mass customization development asks companies to rapidly adapt their MS and, more generally, their supply chain. This is why the success of implementing the leagile strategy helps to switch from one production strategy to another by increasing the flexibility in changing the capacity when needed with minimum impact and developing a resilient supply chain that anticipates disruption. This is the case of RMSs that enable seamless transition between different production strategies by enhancing flexibility in capacity adjustments. Thus, the production strategy and capacity investment will shift from the strategic to the strategic-tactical level.

Similarly, the leagile strategy has changed the paradigm of building and managing the warehouse. This imposes the development of new business models. For instance, outsourcing the storage allows dynamic warehouse size planning as a function of several factors. Hence, this decision is shifting from a strategic towards a more tactical level, more

preciseley a strategic-tactical level as some resistance still exist from industrial companies to switch to such new business models. The main reasons are the lack of maturity of business models and unwillingness to create a strong dependency on an external company for a critical process.

The leagile strategy also impacts both the sizing and capacity allocation decision problems. For the sizing problem, the leagile strategy changes the decision nature but not the time horizon, as we still dealt with at a strategic level. It introduces a new option involving the rental decision in addition to the investment decision for the number of vehicles. Regarding the capacity allocation problem, the leagile strategy does not affect the time horizon, but the CIAOC properties. This problem remains at the operational level, but the leagile strategy introduces new constraints aimed at improving agility. These constraints may include new resource capabilities and the integration of new communication protocols.

Figure 3.12 illustrated all these discussed changes on decision problems due to the impact of leagile strategy.

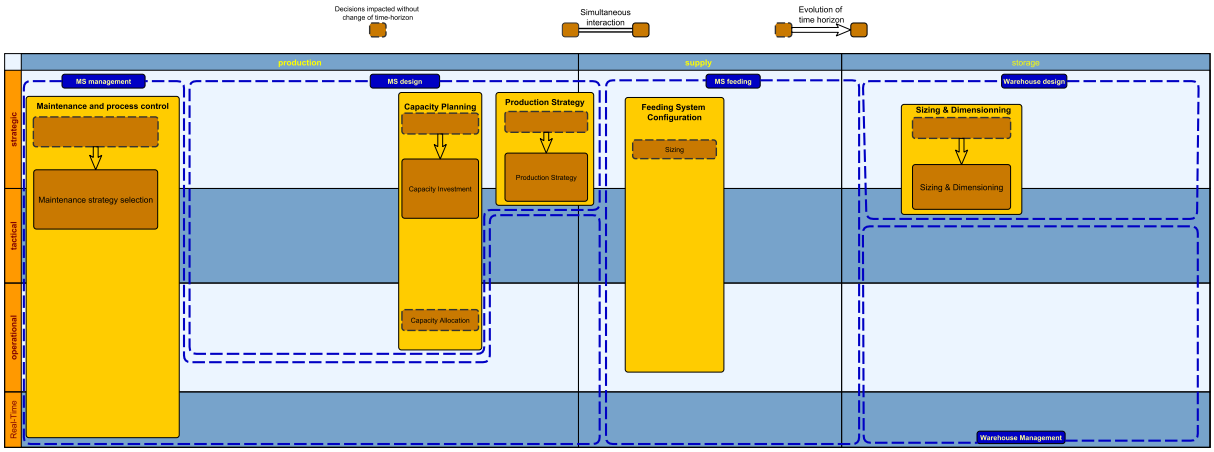

Figure 3.12: The impacts of leagile strategy on time horizon of decisions.

Human-centred MS

The first example of decision whose time horizon is impacted by the Human-centered MS transformation driver is maintenance capacity planning. When dealing with this decision of the production activity, the skills and number of resources to meet the demand for maintenance work must be determined. Moving towards a more human-centered MS involves considering new job skills and promoting lifelong learning to develop worker polyvalence. As a consequence, the capacity planning must be reviewed on a short-time basis as the worker skills evolve continuously.

For similar reasons, especially regarding worker heterogeneity, when deciding on line configuration, we expect that the line balancing and equipment selection will be considered

at the strategic-tactical level.

The human-centered MS driver also impacts the line feeding policy. The number of drivers per delivery is commonly taken into account as a constraint for deciding the line feeding policy. However, due to the dynamicity of an MS within Industry 4.0, the number of drivers will vary each day, asking to review the policy frequently. Hence, we expect the line feeding policy to shift at the operational level.

Picking and routing strategies are also being impacted by the human-centered MS driver. Order picking is highly repetitive, and it is required to consider human factors such as perception and cognition. Furthermore, the full automation of the picking systems in practice is still limited due to lack of flexibility and adaptability. Therefore, supportive and substitute technologies are often invoked when discussing the level of automation in order picking systems. More human-centered MSs will undoubtedly lead to supportive solutions where intelligent systems assist order pickers. As a result, changing the picking and routing strategies more frequently is possible without impacting internal logistics performance.

In the same context, the willingness to restore the central role of human operators in MSs asks industrials to develop solutions that guide the order pickers to the correct locations. Based on such solutions, the dynamic version of the Workforce Assignment Problem (WAP), which consists of dynamically assigning the orders to the order pickers, has become possible in practice. Hence, we expect to handle the WAP in real-time rather than at an operational level.

Figure 3.13 illustrates the impacts discussed on decisions due to the Human-centred MS driver.

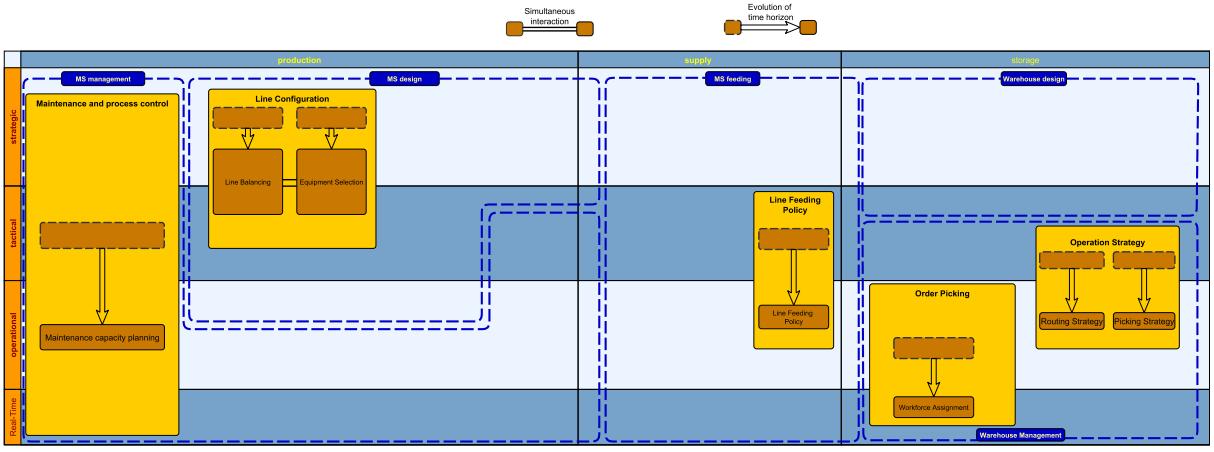

Figure 3.13: The impacts of human-centered MS on time horizon of decisions.

Adaptable MS

It is possible that the evolution of the time horizon of one decision can be caused by multiple drivers not only one. As we did before, in this paragraph, we will also concentrate only on the impact of the transformation driver: adaptable MS, excluding the effects of other drivers.

The Choice of Type of Line is one the problem discussed here. The frequency of this decision remains more than one year, as explained in Subsection 2.4.1.2, hence the time horizon is not affected. However, industrial companies are aware of the necessity of reconfigurable lines to face Industry 4.0 challenges, and of the consideration of new criteria, such as reconfigurability and uncertainty, when deciding the type of line.

Also impacted by the adoption of adaptable MS are Line Reconfiguration Problems. The necessity to rebalance the line is mainly due to many sources of uncertainty, especially concerning the demand, that can occur in an Industry 4.0 environment. As a result, the line rebalancing problem can no longer be considered at the tactical level. Instead it must be dealt with at the operational level. For the same reasons, line relayout decisions must be tackled on a short-term basis.

Besides, the adaptable MS also impact on the routing problem when dealing with line feeding. With the increased adoption of mass customization, many models are produced, making the supply activity more complex to handle, because of the limited stocking space at the border of the line, and the difficulty to store all components continuously. For instance, the routing considers different models and number of vehicles as inputs to define the optimal sequence of stations to be visited. Moreover, within Industry 4.0, the market volatility imposes to switch from one model to another on a daily basis. Thus, solving the routing as a tactical decision does not make sense anymore. Instead, its consideration at the operational level and the necessity of connecting the cyber and physical levels must be highlighted. Furthermore, methods promoting collaboration among vehicles are needed.

The adaptable MS transformation driver affects several decision problems belonging to the storage activity. For instance, the time horizon of the material handling selection and the configuration of AS/RS decisions does not change. However, the two problems are impacted by the multitude of criteria with quantitative and qualitative attributes. For the configuration of AS/RS, the exponential number of components to handle to produce several models introduces storage areas with different inventory turnover and makes analytical approaches no longer adapted for estimating the travel time.

Similarly, the block stacking problem is impacted by the adaptable MS. The flow of components in the warehouse becomes highly dynamic, asking for making the block stacking decisions to be taken more frequently. Existing approaches to solve the block stacking problem are no longer adapted because they consider uniform product flows with specific quantities in each period. Furthermore, new business models offer the possibility of leasing extra warehouse space to increase or decrease the capacity of warehouse according

to flow fluctuations.

Lastly, as discussed before, the configuration of zones becomes dynamic due to the factors impacting (e.g., demand, storage strategy, and storage layout). To include these factors and predict the optimal zone size, ML algorithms are increasingly resorted to, thus providing the possibility to consider the zoning size problem at the operational level.

Discussed changes resulting from the impacts of the adaptable MS are depicted in Figure 3.14.

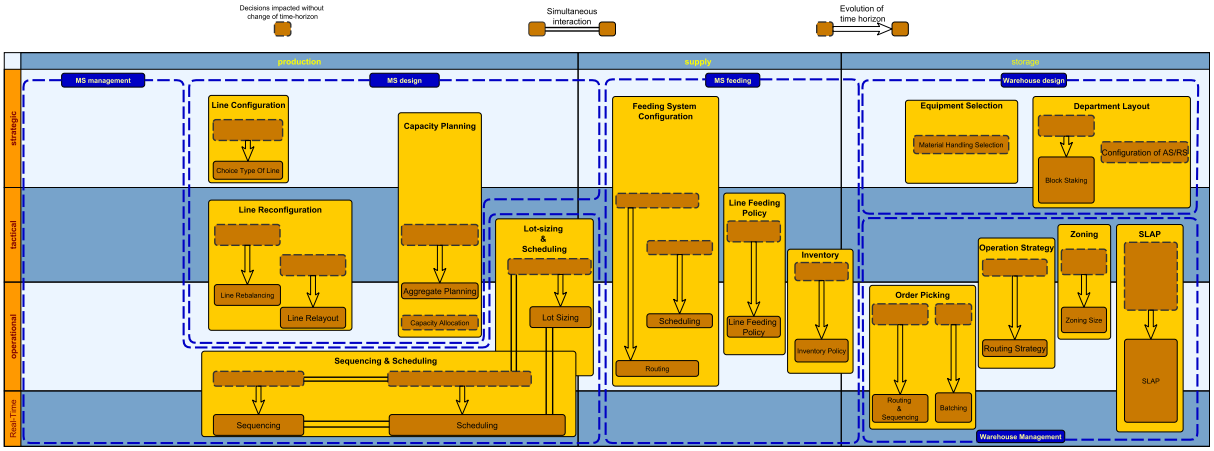

Figure 3.14: The impacts of adaptable MS on time horizon of decisions.

Real-time monitoring

Real-time monitoring accelerates decision-making and allows acting more rapidly on an MS. For instance, we improve the maintenance control can be improved by equipping the product/resource with sensors to upgrade them to smart products/resources able to communicate their state and generate alarms when maintenance is required.

Inventory planning & control and loading, in the supply activity, generally require near real-time operations in order to achieve process optimization. More specifically, the information about the inventory level of each component in the value chain must be updated very frequently, particularly in the warehouse and supermarkets. The need to continuously tracking the inventory is leading to the creation of new policies and methods for inventory management. Furthermore, Inventory Management System (IMS) solutions started emerging as independents module interfaced with the ERP (Enterprise Resource Planning) to meet this need. The ERP is a software solution that integrates and manages core business processes like finance, human resources, supply chain, manufacturing, and sales. It consists of customizable modules or applications that meet to specific organizational needs and can be accessed by employees across different departments. The primary purpose of ERP is to streamline and automate operations, improving overall efficiency and providing real-time visibility into business activities.

Note that to optimize the loading problem and decide which component must be picked in each trip by the transporter, we need to make an IMS interoperable with ERP and WMS(Warehouse Management System) for stock locating. The WMS plays a crucial role in the enterprise by controlling inventory locations and enhancing accuracy and traceability throughout the warehouse processes. It also enables data-driven decision-making for overall warehouse performance optimization. For instance, it optimizes picking routes, suggests appropriate storage locations, and assists in workforce assignment. However, in the context of Industry 4.0, the WMS needs to provide the ability to deal with the routing & sequencing and the workforce assignment in real-time. To achieve this, it must capture data on product movements, order statuses, and inventory transactions. Therefore, real-time monitoring asks solution providers of WMS to enhance the existing modules of order picking, packing, and wave planning.

Figure 3.15 shows based on the same convention of Figure 3.9, the discussed changes. A problem whose time horizon is being shifted is represented by two brown boxes, one dotted, the other continuous, linked by an arrow to show the shift.

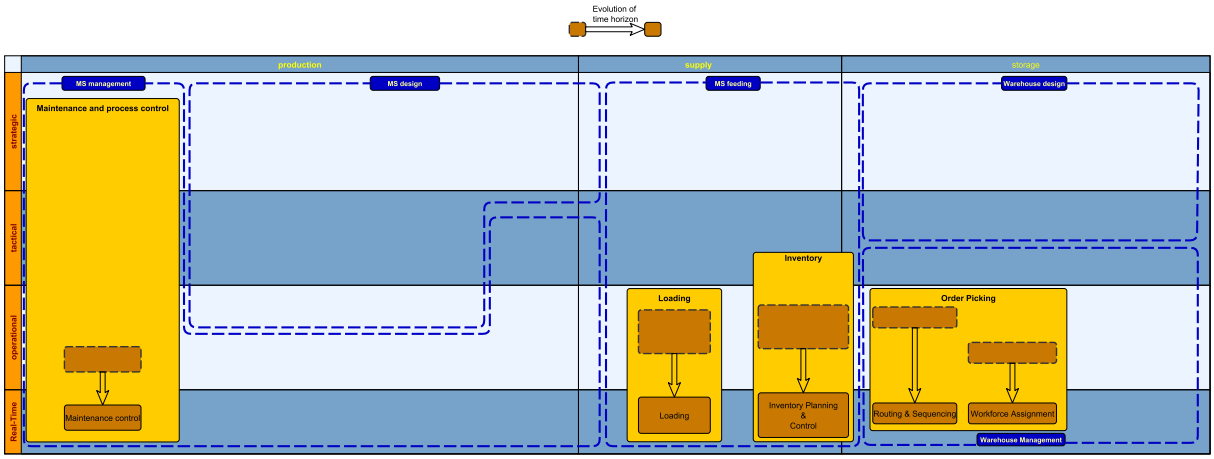

Figure 3.15: The impacts of real-time monitoring on time horizon of decisions.

Decentralized control approach

Subsection 2.4.2 pointed out the evolution of MS types in the context of Industry 4.0. More specifically, for RMSs we deduced the necessity to add a controllability feature to make the RMS controller decentralized, so as to have self-configurable and auto-adaptive MSs. By doing so, deciding the line relayout problem at the operational level becomes possible.

A decentralized control approach can impact certain problem families, e.g., sequencing & scheduling problems. On the one hand, adopting a classical centralized approach such as manufacturing planning and control to solve these problems is not recommended due to a lack of responsiveness. On the other hand, the complete decentralized control approach

cannot guarantee to have a globally optimal solution, but provides the agility required. Nowadays, we tend to use a hybrid approach, i.e., adopting a gradually decentralized control approach, i.e., that distributes decision-making among nodes. For instance, the decision concerning the sequencing & scheduling is taken in the ERP node at the work center level on a daily basis. But, this decision is refined in the MES (Manufacturing Executive System) node at the workstation level by considering the actual information coming from the physical world for orchestrating the production execution and dispatching jobs dynamically. For these reasons, we expect that using a decentralized control approach will lead to managing the sequencing & scheduling decision problem in real-time rather than at an operational level.

For the supply activity in MS, some studies in the literature focus on the static scheduling approach, which asks for the material requirement to be known several days before its consumption time on the workstations. However, the mass customization imposed by the industry 4.0 environment, prevents from adopting such an approach. When dealing with the feeding system configuration, the scheduling decisions must be adjusted according to the customer demand (sales order registered in ERP) to deliver the necessary parts at the right time. Also, the labor management system (LMS) and MES must obtain the equipment state and production execution status, respectively in near real-time. This approach is called *dynamic* or *online scheduling* and leads to considering the scheduling problem at the operational level. The WMS will be responsible for gathering information from ERP (e.g., manufacturing order status), LMS(e.g, manual picking orders status), and MES (e.g., task completion status), then, relay it to each tow train and letting it decide its trip in coordination with other tow trains. To put it differently, depending on the progress of the task execution, and the manufacturing orders released, as well as the preparation made of the picking orders, each tow train will be able to decide whether to go into a specific workstation to retrieve products and transfer them to other locations, or to go for picking the required components from warehouse to feed the workstations. The idea is to enable better coordination and synchronization between warehouse operations and manufacturing processes, improving efficiency and productivity. Thus, we expect that the decentralized control approach will promote dynamic scheduling.

The routing strategy problem in the storage activity is equally impacted. Introducing Autonomous Mobile Robots (AMRs) for dealing with intralogistics problems involve other issues such as collision management and energy consumption. Therefore, solving the routing strategy requires a solution allowing to react dynamically to changes and find a collision-free route with lowest possible energy. Using a centralized approach, as is the case for AGV, is not adaptable because a central unit takes control of routing decisions and suppose a static environment. However, allowing each node to decide its path according to the traffic based on updated information will allow dynamically handling of the system

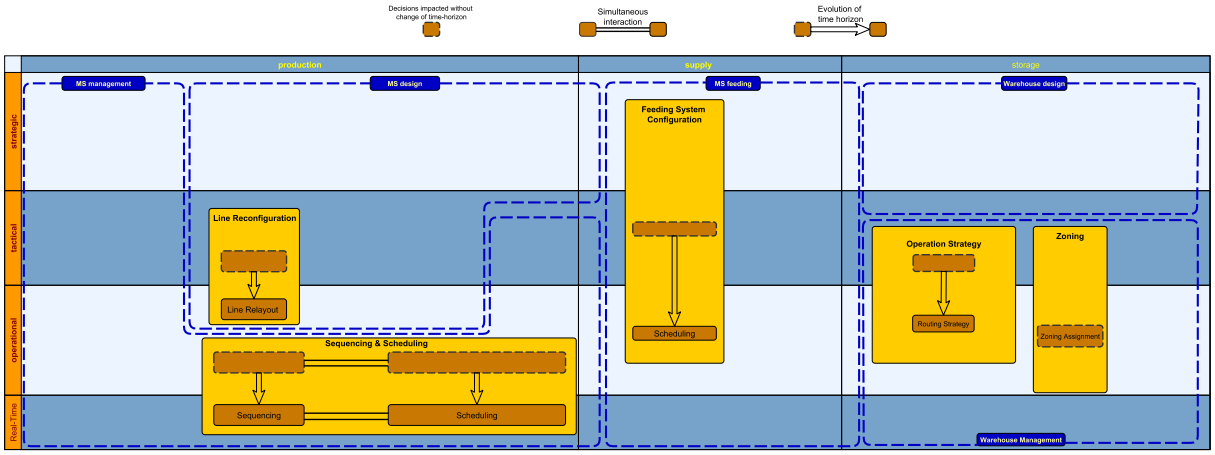

Figure 3.16: The impacts of decentralization control approach on time horizon of decisions.

state. Furthermore, it will reduce the complexity of the problem, even if the obtained routing path is likely to be sub-optimal. Hence, using a decentralized control approach, the routing strategy can be considered at the operational level.

Based on our analysis, the time horizon of the zoning assignment problem (ZAP) will not change. However, the solution approach to use is impacted. Centralized approaches are unsuitable because the configuration of zones becomes dynamic, i.e., zone could added, removed or divided dynamically by varying the picking locations. Therefore, the ZAP is still dealt with at the operational level but decentralized control approaches are to be preferred.

All these discussed changes are illustrated in Figure 3.16. The decentralized control approach impacts the ZAP, not on the time horizon, but on the decision nature itself, e.g., by including the dynamic aspect in this problem. The cartography tool allows us to illustrate such an impact on one decision due to one transformation driver even if no change in the time horizon occurred. Graphically, the brown box illustrating a decision problem is kept at the same time horizon level as defined in the cartography (Figure 3.17), and only its contour is drawn with a dashed line to point out the decision evolution. This evolution can be due to changes in some of the CIAOC properties or in the decision nature itself, like the dynamic aspect included in the ZAP.

Using KPI 4.0

Among the decisions that are most impacted by this tranformation driver, there are those concerning the Line Configuration, i.e., Line Balancing and Equipment Selection, which have historically been strategic problems. Mainly, the ergonomic and sustainability aspect discussed in Subsection 2.3.3.5 are more and more considered criteria to take into account when dealing with such decisions. With future development, these problems will shift to a more strategic-tactical level because energy price, energy consumption of

machines, and the evolutions of ergonomic solutions will be subject to faster and faster changes. Hence requiring the adaptation of the production line in the medium term to remain competitive.

Similarly, Fcaility Layout decisions will be more and more frequent , due to the agility aspect, thus inducing short-term changes in plant configuration. Mass customization and short PLC will further contribute to this effect. Therefore, the facility layout will shift at a more strategic-tactical level.

As already explained in Subsection 2.3.3.5, the loading problem within Industry 4.0 will have to take energy consumption into account in order to provide the supply plan. In other words, the loading decision will be taken according to energy price variations throughout the day, the real-time autonomy of tow trains, and the current location of items that must be loaded in each slot. Therefore, we expect the loading decision to move to the real-time horizon.

Finally, the Location Planning of Feeding System (LPFS) and the storage equipment selection do not change as regards the time horizon, and both remain at the strategic level. However, the criteria for making these decisions will increasingly include sustainability and agility.

Impacts discussed on decisions due to the use of KPI 4.0 driver are depicted in Figure 3.17.

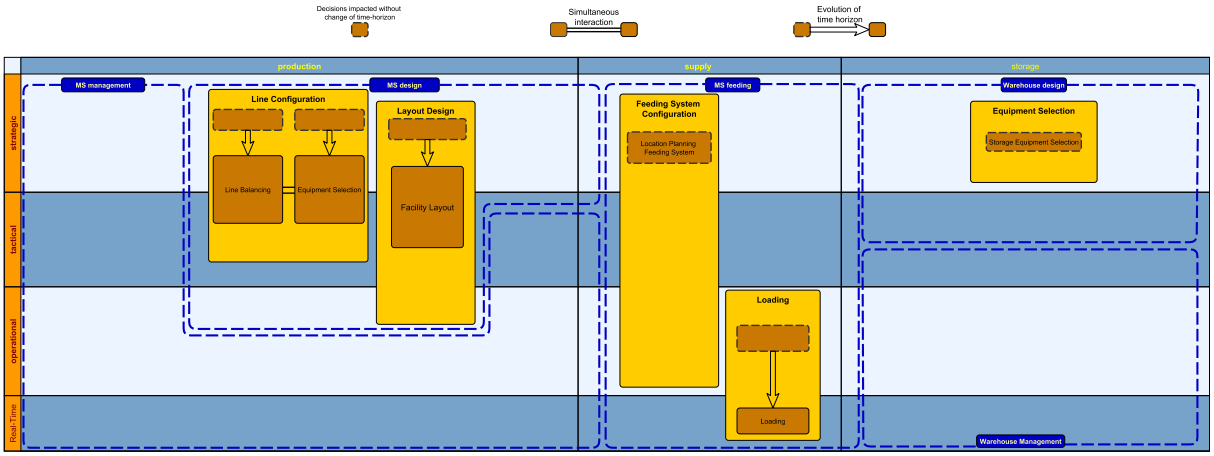

Figure 3.17: The impacts of using KPI 4.0 on time horizon of decisions.

3.7.2 Evolution of Decision Interactions

Based on decision interactions identified and the literature review, this section reviews some notable examples of interactions among decision problems, which have evolved due to Industry 4.0.

The modularity and convertibility characteristics offered by RMS led ALBP and line layout problems to be closer. As a result, the interactions identified and discussed previously between the two decisions can evolve, allowing them to be adressed jointly in the new context of industry 4.0. Furthermore, due to the VUCA world, including the MS reconfiguration in the design phase is required. Thus, the integration of the line layout problem with the line relayout problem is also expected to become possible soon.

Supermarket locations affect the size of vehicles used to feed component lines. Therefore, if the supermarket location can be determined in each period, then the number of previously decided vehicles must be reviewed. Since it is an investment decision, companies must seek the integration of the two decisions by considering uncertainty over periods. By adapting a leagile strategy, this investment can be switched into a rental decision by resorting to a new form of acquisition of vehicles, e.g., loaning only for periods of usage peak.

The work of Scholz and Wäscher (2017) jointly tackles the batching, and routing and sequencing decisions, showing the benefits of using an integrated approach, especially in the case of multi-block layouts. Recently, Kübler et al. (2020) propose to jointly solve the SLAP, batching, and routing and sequencing problems in a dynamic environment. The authors propose a sequential iterative method to be able to tackle large-scale problems.

Recently, Nourmohammadi et al. (2018) solve the ALBP and LPFS jointly, under the assumption of uncertain demand and stochastic task time. The authors propose a model and test it to on a real case study. Battini et al. (2016) try to integrate the ALBP with the sizing problem. They formulate a model to minimize the required assembly and supermarket operators, having a fixed cycle time. Silva et al. (2020) integrate three decision problems: workforce assignment, routing & sequencing, and batching. They formulate and solve the integrated problem in a real-life spare parts warehouse. Emde and Schneider (2018) solve the sizing and routing problems jointly, and Emde and Boysen (2011) integrate the routing and scheduling problems. In the future, using the potential of the new techniques and algorithms developed in fuzzy logic, game theory, ML, and reinforcement learning, the three decisions will likely be tackled together. Following the same logic, one of the possibilities to investigate is to address the three problems of routing, scheduling, and loading simultaneously, since Fathi et al. (2015) already solved the scheduling and loading problems jointly.

The proposed cartography tool can also help illustrating these evolutions of the interactions among decisions in MS. This is shown in Figure 3.18 and Figure 3.19. We use red and green colors to distinguish between new and potential simultaneous interactions, respectively. Potential simultaneous interactions among decisions mean that no work exists that integrates these decisions as far as we know. However, based on our analysis of the industry 4.0 scenario, we expect their integration in the future.

Figure 3.18 illustrates also decision problems that are tackled in sequential way, and

may be interested to deal with in simultaneous way in the context of Industry 4.0, as explained before and summarized below:

- the ALBP and line layout problems can be addressed jointly.
- the line layout and the line relayout problems can be integrated.
- the LPFS and sizing must shift to simultaneous.

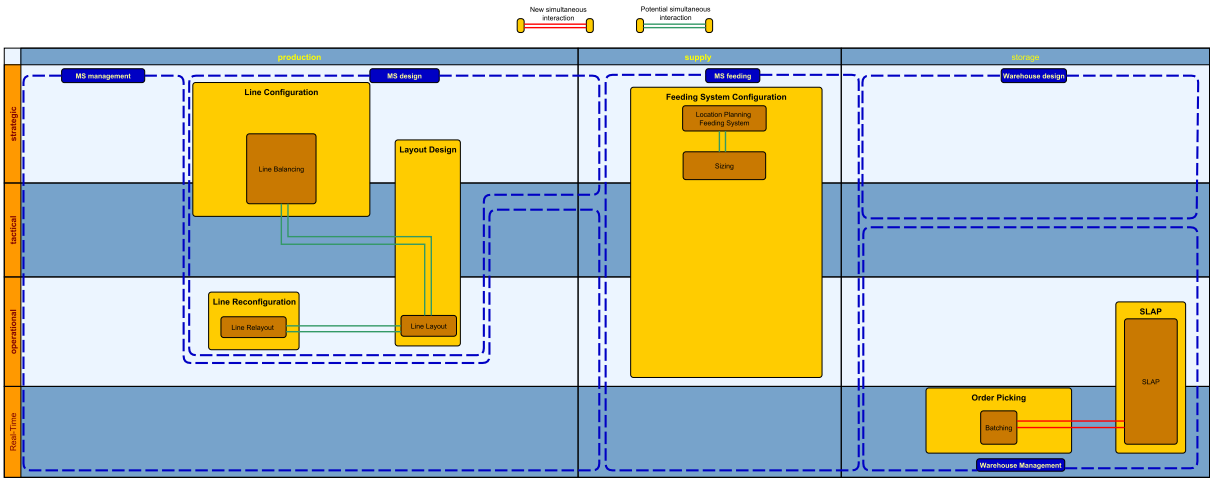

Figure 3.18: Illustration of some evolution from sequential to simultaneous interactions.

Figure 3.19 also uses the cartography tool to illustrate the aforementioned evolutions. The meaning of red and green colors remains the same. Moreover, we create a new type of interaction to represent the integration of more than two decisions. Two lines still refer to simultaneous interaction. We only added a diamond to group the two lines coming from the different decisions.

The new simultaneous interactions of decisions imposed by the Industry 4.0 environment already discussed are shown in Figure 3.19, and listed below:

- the solving of the ALBP and LPFS jointly.
- the integration of the ALBP with the sizing problem.
- the integration of the workforce assignment, routing & sequencing, and batching problems.

Finally, Figure 3.19 shows the potential simultaneous interactions of decisions expected to take place in the new industrial revolution, and a brief description about them is given in the following:

- the ALBP has been tackled with line layout jointly, and recently, it was integrated with the sizing problem. Dealing with the three problems will optimize material flow and vehicle utilization while enhancing the implementation of just-in-time delivery in line feeding operations.

- the same goes for the sizing, the routing, and the scheduling problems. Until now, they were tackled pair-wise, but, based on their structure, it is possible to envisage their integration.

- the future research may tackle the routing, scheduling, and loading problems together.

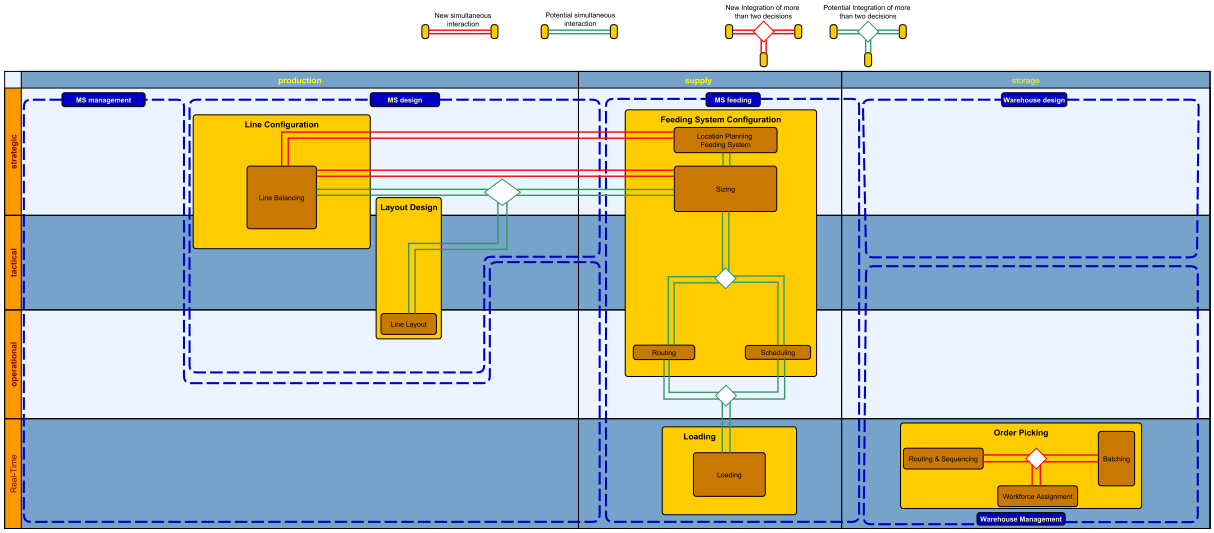

Figure 3.19: Illustration of new and potential integration of decisions.

3.8 Conclusion

In this chapter, we proposed a novel three-step procedure to study the decision problems in MSs and their interactions.

First, we provided a taxonomy of identified decision problems in MSs according to four factors: the time horizon, the problem family, the problem branch, and the activity type.

Secondly, we introduced the concept of decision properties: constraints, inputs, assumptions, outputs, and criteria (CIAOC). The conceptual model of the decision view of an MS highlights these properties and describes how they are related to the decision.

Later, based on the conceptual model of the information view of an MS, we defined the MS features and classified them according to six classes: Process, Product, Resource, and External effet, Cost, and Time (P_sP_tRE-CT). Finally, we codified the MS features and

used them to describe the CIAOC properties of each decision. The analysis of decision view lead to consider new constructs. Furthermore, using the CIAOC properties allows to have a generic approach for modeling decisions adapted for both the old and new scenarios, i.e., before and during Industry 4.0 revolution.

Thirdly, the modeling of decision interactions provides a logical decision process that can be used when designing an MS. Besides, we provided an analysis of MS features based on the old scenario, and we pointed out some evolutions either on constraints, or criteria to cope with the evolution of decisions within industry 4.0.

Thirdly, to understand how decisions impact each other and promote the development of integrated models, we proposed to study their interactions. Based on the literature, we have identified three types of interactions, correlated, sequential, and simultaneous. In our analysis, we discovered that the MS type might affect the interaction type in some situations. As a result, we introduced two new interactions, DMS-only sequential and FMS/RMS-only sequential. Finally, to simplify the analysis of the interaction type among decision problems, we defined a set of rules based on the CIAOC properties and the literature.

Then, we developed a novel cartography tool to summarise both the taxonomy and the interaction of decision problems. It can be helpful for researchers and practitioners in identifying synergies between decision problems belonging to different activities, in developing new integrated and robust models, and thus represent a valuable contribution. Our analysis of Industry 4.0 proves that huge efforts are currently being made to integrate decision at different levels of different activities. This means that more complex problem will be adressed, requiring the development of new solution approaches.

Lastly, using the proposed cartography tool, the impact of the main transformation drivers (identified in Section 2.3) on the decision evolutions, discussed in Section 2.4, can be visualized. Our analysis of the impacts of drivers of the decision evolution highlights the necessity to change the way of modeling the enterprise and lead us to ask several questions:

- How Enterprise Modeling (EM) is evolving, and how EM methodologies must evolve?
- Can we use the main transformation drivers of Industry 4.0 to define the new principles of EM required by the Industry 4.0 environment?
- Are the defined views sufficient to cover all enterprise aspects? If not, which are the other views needed?

The goal of the next chapter is to answer these questions and propose a new, suitable EM methodology, capable to address Industry 4.0 requirements.

Chapter 4

New Methodology of Enterprise Modeling Oriented Industry 4.0

Résumé

Ce chapitre se concentre sur la création d'une nouvelle méthodologie de modélisation d'entreprise (MEMO I4.0) adaptée à l'environnement de l'Industrie 4.0. L'introduction expose le besoin de modèles pour structurer les décisions lors de la mise en place de nouvelles usines, et comment la modélisation d'entreprise fournit un cadre structuré pour obtenir des modèles cohérents. Cependant, le paradigme de prise de décision dans les systèmes de production a changé avec l'Industrie 4.0, ce qui exige une évolution aussi des méthodologies de modélisation d'entreprise. Ce chapitre aborde ensuite les nouvelles approches de modélisation pour traiter la généricité des modèles dans l'Industrie 4.0. Il définit ensuite la nouvelle méthodologie MEMO I4.0, décrivant l'ensemble du processus de modélisation et visant à réduire le temps de développement des modèles. La méthodologie est dérivée des principes clés identifiés pour faire face aux défis de l'Industrie 4.0. Elle comprend des objectifs de modélisation associés à des dimensions incorporées dans le cadre MEMO I4.0. Le chapitre propose également une feuille de route pour appliquer MEMO I4.0 pour développer des modules à l'étape stratégique/tactique, en utilisant un exemple d'entreprise fictive (E1) pour expliquer les concepts de la méthodologie en détail. Il aborde l'application de MEMO I4.0 à l'étape opérationnelle, c'est-à-dire dans des cas d'utilisation spécifiques basés sur les modules développés. En conclusion, ce chapitre présente la méthodologie MEMO I4.0, qui respecte les principes et les dimensions nécessaires à l'Industrie 4.0. Il souligne l'importance de prendre en compte la généricité et l'échelle dès le début du processus de modélisation pour obtenir un modèle cohérent et modulaire. La méthodologie est conçue pour offrir agilité et flexibilité dans le processus de modélisation, permettant ainsi une évolution plus facile des modèles développés. Enfin, ce chapitre prépare le terrain pour l'application de MEMO I4.0 dans un cas réel, qui sera abordée dans le prochain chapitre.

Contents

4.1 Introduction 140
4.2 New Modeling Approaches for Enterprise Modeling in Industry 4.0 142
4.3 Methodology of Enterprise Modeling Oriented I4.0 (MEMO I4.0) 143
4.3.1 Key Principles of MEMO I4.0 144
4.3.2 Modeling Objectives of MEMO I4.0 145
4.3.3 Dimensions of MEMO I4.0 148
4.3.4 Specifications and Purpose of MEMO I4.0 151
4.4 Application of MEMO I4.0: Strategical/Tactical Stage 153
4.4.1 Definition Phase: Requirements and Specifications of Modeling 155
4.4.2 Design Phase: Instantiating Metamodels of Views . . . 165
4.4.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module 169
4.4.4 Maintenance Phase: Versioning of the Model of Each Module and Generation of the Corresponding Documentation 177
4.5 Application of MEMO I4.0: Operational Stage 178
4.5.1 Definition Phase 178
4.5.2 Design phase 178
4.5.3 Implementation phase 179
4.5.4 Maintenance Phase 180
4.6 Conclusion 180

4.1 Introduction

The implementation of a new plant is based on several decisions at different levels concerning different activities (see Chapters 2 and 3). We resort to models to structure such decisions, optimize them, and simulate the impact of their possible implementation. The role of EM methodologies is to offer a structured framework to obtain consistent models. In the literature, several EM methodologies have been proposed to make the modeling process consistent and straightforward, so as to facilitate the study of MS behavior (see Subsections 1.3.1 to 1.3.3).

In a similar way, implementing a new smart factory requires making decisions and needs models to study the behavior of the MS and optimize its configuration. However, as discussed in the previous chapters, the decision-making paradigm in MSs is changing within Industry 4.0. Hence, even MS modeling must evolve either by adapting existing EM methodologies or resorting to new ones. Our literature review showed that classical EM methodologies (e.g., GIM, CIMOSA) offer only a part of the capabilities required to face the challenges of Industry 4.0, while the new EM methodologies (e.g., RAMI 4.0, IIRA) lack the possibility of a complete conceptual-to-implementation workflow. Therefore, neither the former, nor the latter are today adapted to face all identified

challenges of Industry 4.0, hence new EM methodologies are needed, as discussed in Section 3.8. This chapter aims to propose such a new EM methodology and answer the third research question of Section 1.5, which deals with how EM can be enhanced to address the decision-making challenges arising from Industry 4.0 and subsequently develop Industry 4.0-oriented Decision-Support Systems.

Of course, building an EM methodology must consider the status quo, as explained in Chapter 3 and shown in Figure 4.1. Mainly, three elements of discussion must be retained from this figure. First, the smart factory imposes several challenges, and we presented the main transformation drivers to face each challenge (see Table 2.1). Second, these transformation drivers impact decisions on either the time horizon, the properties, or the nature as explained in Section 2.4 and subsection 3.7.1. Third, there is a necessity to develop DSSs to make one or several decisions in complex environments (see Subsection 1.4.1).

In the following of this Section, Section 4.2 presents new modeling approaches for dealing with the genericity of models in order to apply enterprise modeling concepts within Industry 4.0. Next, Section 4.3 defines a new EM methodology (MEMO I4.0) that describes the entire modeling process and aims to reduce the development time of models by avoiding starting from scratch each time a new use case is at the study. The section starts with the transformation drivers identified in Section 2.3 to face the challenges imposed by Industry 4.0 in order to define the key principles of MEMO I4.0, before deriving the modeling objectives of MEMO I4.0 and associating with each modeling objective a dimension that must be incorporated into the MEMO I4.0 framework. Section 4.4 defines a roadmap to apply MEMO I4.0 for developing modules;A fictitious enterprise is used as an example to explain in detail the concepts of the methodology. Section 4.5 deals with the application of MEMO I4.0 at operational stage in some specific case studies based on modules developed. Finally, Section 4.6 draws the conclusions of the introduction and application of MEMO I4.0.

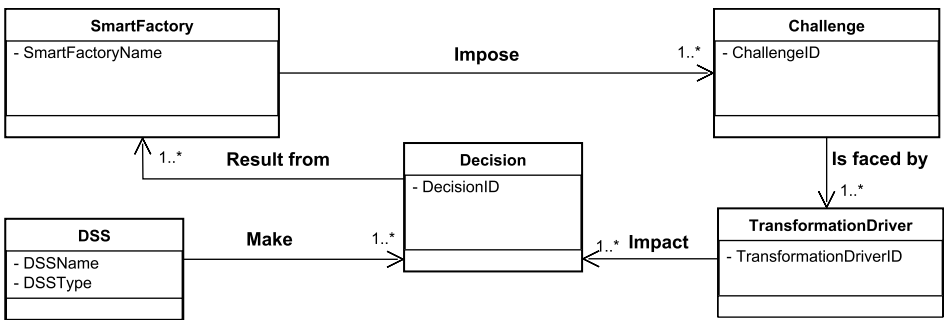

Figure 4.1: UML diagram of the smart factory

4.2 New Modeling Approaches for Enterprise Modeling in Industry 4.0

During the modeling phases, one of the first steps is to choose the suitable genericity degree of the models. To do so, it is first required to consider the existing enterprise guidelines that impose specific modeling choices: indeed, the enterprise vision might determine the potential evolution of current practices and some future system targets that must also be considered during the design phase. After choosing the genericity degree according to the expected modeling context, we need to consider the possible modeling choices that allow to achieve this degree.

Once the genericity degree is chosen, three main modeling approaches can be adopted, namely:

- The independent approach considers the systems to model independently. An n-switch mechanism is used to select the model corresponding to each system.
- The intermediate approach captures common features among the systems. It uses the same concepts to model them and adopt the n-switch mechanism for features that are not shared.
- The integrated approach consists in modeling common features using the same concepts, and analyse how we could use a generic concept even for features that are not shared to map them.

A first example applying the independent approach deals with modeling the line feeding policies, including line stocking, kanban, sequencing, and kitting. It emphasizes the need to have a specific model for each policy due to their distinct trigger conditions and complexity, and using a discriminant variable as a switch between the different policies in the model. A second example applying the intermediate approach deals with modeling storage strategies, namely dedicated and shared. The example highlights the common features such as item reception, control, assignment, and storage process, while the decision of storage location differs between the two strategies, and thus requires a specific treatment based on the storage strategy of an item. A third example applying the integrated approach concerns the modeling of three production policies (push, pull, and hybrid). For this case, the integrated approach is based on the common features of receiving and sending signals between workstations to trigger production. We propose to use the material flow as a signal for modeling the push policy, allowing for a generic model that can be switched between different production policies. Appendices D.1 to D.3 provide a more in-depth description of the aforementioned examples.

4.3 Methodology of Enterprise Modeling Oriented I4.0 (MEMO I4.0)

As explained in Subsection 1.4.2, we resort to modeling to study the behavior of enterprise systems. The enterprise model results from the composition of several submodels. Building submodels and ensuring their consistency requires respecting EM principles and having a reference framework. Usually, in the literature, dimensions are used to describe EM frameworks and achieve the modeling objectives. An EM methodology seeks to define the modeling principles and the overall framework, i.e., dimensions and their roles in modeling for succeeding in the modeling process and deriving a set of models that constitute the enterprise model.

Nowadays, VUCA world characterizes the MS context in Industry 4.0. Several factors (internal and external) act on MS, impose constraints on it and subsequently force it to evolve. Since this manuscript focuses on MSs as an enterprise object in the context of Industry 4.0, this section presents MEMO I4.0, a new EM methodology adapted for modeling smart factories. We will therefore explain its principles and framework, how it is adapted to Industry 4.0 and how to use it to develop models that represent the structure and behavior of the business entity.

Figure 4.2 summarizes how EM methodologies are structured, a guideline we will be following in the definition of MEMO I4.0.

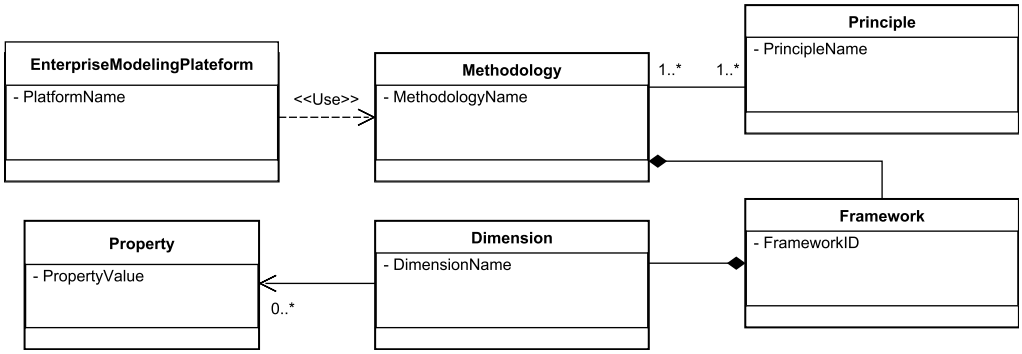

Figure 4.2: UML diagram of the EM methodology.

The ultimate goal of MEMO I4.0 is to obtain a Digital Twin (DT) of an MS, in such a way that models composing such DT are developed to be scalable, and also reconfigurable using near real-time data (El Alaoui El Abdellaoui et al., 2020). Therefore, MEMO I4.0 is a fully Data-Driven EM approach featuring the Industry 4.0-oriented characteristics of Table 1.2. However, MEMO I4.0 stands as a new, smart EM tool meant for companies with a long-term vision of future orientations, which operate in a structured, standardized framework based on their DNA and guidelines.

In the following, we need to distinguish four notions: transformation drivers, principles, objectives, and dimensions. These notions will allow us to review the existing principles and frameworks, and derive those of MEMO I4.0. Based on the main transformation drivers presented in Section 2.3, the required key principles of MEMO I4.0 can be derived. Then, each key principle can be associated with a specific modeling objective. Lastly, each modeling objective of MEMO I4.0 characterizes one dimension of its framework.

According to our vision, we classify three among the six principles discussed in Subsection 1.3.1 as modeling objectives of MEMO I4.0, namely the choice of type of synchronization, the processes versus agents, and the adaptation of models to the context, i.e., the modeling level. Furthermore, some of the most-known frameworks did (e.g. GIM or CIMOSA), we consider the view as a dimension of MEMO I4.0 rather than a principle. As regards the two remaining principles (plural nature of the enterprise model; decoupling of the three fundamental types of flow), we aggregate them in only one principle, called *modularity*. This latter allows for building an integrated model of modules, module by module and ensures the decoupling of flows (material, information, and control).

4.3.1 Key Principles of MEMO I4.0

The key principles of MEMO I4.0 are extracted from the main transformation drivers (identified in Table 2.1) of the evolution from the old scenario to the Industry 4.0 scenario as illustrated in Figure 4.3. These key principles are suitable for MEMO I4.0 in the context of a smart factory.



Figure 4.3: EM Principles derived from the main transformations drivers.

4.3.1.1 Modularity Principle

Among the main transformation drivers, the decentralization control approach is compatible with the modularity principle since it allows an incremental approach to developing models. MEMO I4.0 resorts to the modularity principle for increasing the flexibility in building models. Therefore, all models should be made up of modules which are assembled and put in communication among them in order to build the model of the targeted system.

4.3.1.2 Agility Principle

The leagile strategy, using new KPIs 4.0, and new techniques and algorithms are the transformation drivers that impose agility as key principle of MEMO I4.0. In other words, the evolution of business models leading to new system configurations imposes the capability to rapidly reconfigure the models. Besides, new KPIs must be quickly implemented to evaluate the system performance from recently emerged criteria (e.g., ergonomic, agility) and resorting to new techniques and algorithms for making decision even when data is unstructured. MEMO I4.0 uses the agility principle to reduce the development time of models considerably as we will see in Chapter 5.

4.3.1.3 Robustness Principle

The real-time monitoring and the adaptable MS are transformation drivers that can help avoiding pitfalls such as obtaining ad-hoc solutions and seek to develop data-driven models, since when the system evolves, we must adapt the enterprise model to this evolution. Furthermore, these transformation drivers lead to the easy maintainability of models as they are updated according to the system monitoring. However, each time the system model is updated, first, we validate each model that will be integrated into the system separately before running built-in tests to validate the system model. We incorporate this ability into the robustness principle, which allows the validation of modules separately, before integrated tests can be conducted to validate the assembly of modules.

4.3.1.4 Interoperability Principle

Among the main transformation drivers, the standardization of data exchange and the human-centered MS allow to avoid interoperability issues. In terms of modeling, the goal is to develop platform-independent models without an automatic generation but using the expertise of modelers. Therefore, we include these modeling concepts in one key principle called interoperability. In MEMO I4.0 this principle plays an essential role in ensuring the decoupling between models and development platforms to reduce the modeling complexity.

4.3.2 Modeling Objectives of MEMO I4.0

The main motivation of MEMO I4.0 is to reduce the time spent in the development cycle of simulation and optimization projects (modeling, coding, gathering data/information, and verification) in favor of agility, increased reactivity, and process analysis capabilities. Furthermore, it offers a framework to support the modeling process and generate an Optimization/Simulation-Based DSS.

We first propose determining the modeling objectives of MEMO I4.0 related to its key principles. Then, based on these objectives, we will deduce the dimensions of the MEMO

I4.0 framework and ensure that the key principles of MEMO I4.0 are compatible with the dimensions of its framework. In our opinion, each key principle is associated with a specific goal in the modeling process.

4.3.2.1 Modeling Process

The first objective of MEMO I4.0 is to structure the modeling process. Figure 4.4 illustrates the structured approach of MEMO I4.0 that follows the modularity principle and consists of two main stages, organised according to the life cycle phases.

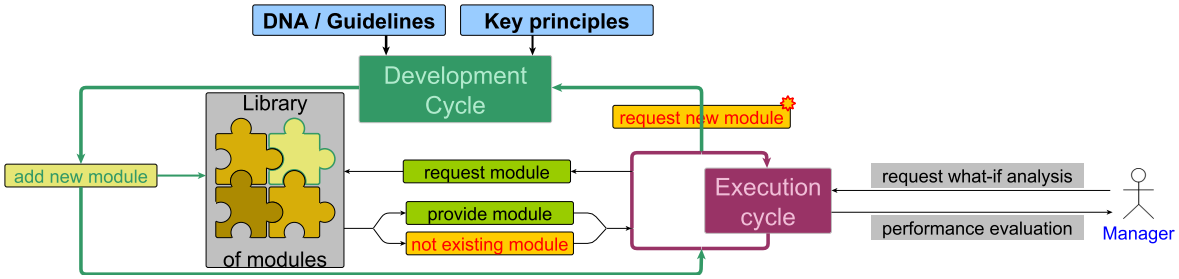

Figure 4.4: The structured approach of MEMO I4.0. The Development Cycle is the key element in building up the Library of Modules.

- At more strategic/tactical stages, the Development Cycle allows to create the modules used at the operational level to build integrated simulation-optimization models based on DNA/guidelines and key principles. Newly conceived modules are added to a Library of Modules to allow for knowledge capitalization;
- During operational phases, the Execution Cycle allows to generate integrated simulation-optimization models and assess the system performances. Missing or obsolete modules trigger a new Development Cycle. Required modules are instantiated with case-specific parameters.

4.3.2.2 Modeling Level

The second objective of MEMO I4.0 is to provide the possibility of adapting the modeling level according to the modeling context. This objective promotes a proactive approach when modeling the concepts, since MEMO I4.0 designs models based on the enterprise guidelines, and vision. With respect to the agility principle, MEMO I4.0 aims to capitalize all efforts made during the modeling so as to obtain reusable models.

4.3.2.3 Modeling Granularity

The third objective of MEMO I4.0 is to ease the tuning of the modeling granularity when the system evolves. Adjusting granularity according to the modeling purposes allows for

quickly validating each module and its integration. A finer granularity allows for a greater flexibility. However, we will be limited in modeling choices and have issues in maintaining developed models if they do not respect the robustness principle.

4.3.2.4 Modeling Consistency

The last objective of MEMO I4.0 is to ensure modeling consistency among different aspects of the system. This is why MEMO I4.0 develops models dedicated for each aspect of the module according to the interoperability principle, i.e., based on a standardized MS feature and using a mapping layer between the business layer and the specific platforms.

Table 4.1 summarizes the analysis of the key principles of MEMO I4.0 by specifying from which main transformation driver(s) each key principle is derived and with which main modeling objective of MEMO I4.0 it is associated.

Table 4.1: Correspondance between MEMO I4.0 principles and modeling objectives.

Main transformation drivers	**Key principles**	**Modeling objectives**
Decentralization control approach	Modularity	Structure the modeling process
Leagile strategy	Agility	Adapt the modeling level
using new KPIs 4.0		
Real-time monitoring	Robustness	Tune the modeling granularity
New techniques and algorithms		
Adptable MS		
Standardization of data exchange	Interoperability	Ensure the modeling consistency
Human-centered MS		

From the point of view of modeling objectives, Table 4.1 can be interpreted as follows: before starting any development of models, we need to have a structured modeling approach, which guides the modeling process. During this process, we decide the modeling level and modeling granularity according to the DNA and guidelines of the company. However, we need to ensure modeling consistency from the early stage of model development to avoid interoperability issues. From the point of view of conceptual modeling, the UML diagram of Figure 4.5 illustrates the link between the transformation drivers, the key principles and the modeling objectives.

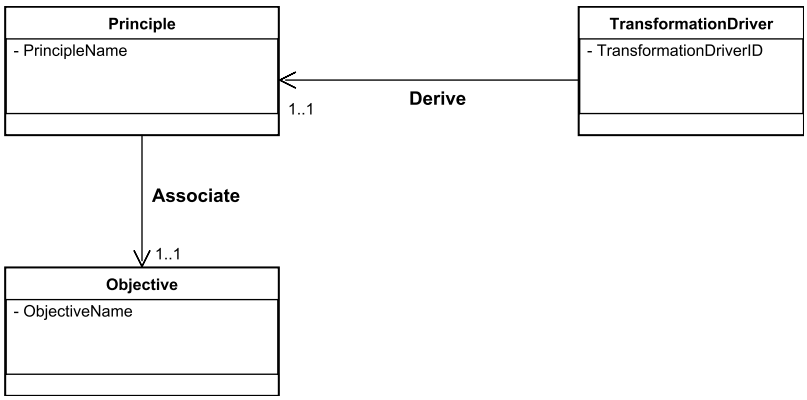

Figure 4.5: MEMO I4.0 modeling objectives, derived from the identified main transformation drivers.

4.3.3 Dimensions of MEMO I4.0

Several EM frameworks have been proposed, and their dimensions discussed. However, previous chapters demonstrate the necessity of the evolution of these dimensions and of their coherence with the key principles. The analysis of the Industry 4.0 scenario conducted so far highlights the necessity of the three integrations: vertical, horizontal, and end-to-end. These integrations are the backbone of the RAMI 4.0 framework that includes three dimensions. The layer and the life cycle value stream play the role of the view and life cycle dimensions in more classical frameworks. The hierarchical level dimension is a new dimension that contributes to building scalable models and avoid to rebuild from scratch models in case of changes. However, these three dimensions seem not sufficient to meet the objective of the adapted modeling level, which aims at reusability of models in different contexts through the genericity dimension.

From the analysis of dimensions of classical and new frameworks, we will try to define the framework of MEMO I4.0. Furthermore, these dimensions must be compatible with its key principles and contribute to achieving its modeling objectives. According to our literature review and comparison of features of the existing EM framework Table 1.2, we propose to exploit the advantages of both classical and new EM frameworks, i.e., to take into account dimensions that cover the context of Industry 4.0 and respond to the challenges of a smart factory. Section 4.4 will delve into greater details about how MEMO I4.0 makes use its dimensions.

4.3.3.1 Life Cycle Dimension

The life cycle dimension of MEMO I4.0 acts as a guideline to structure the modeling process in four phases: definition, design, implementation, and maintenance. At the strategic/tactical stage these phases are followed to build the modules, while at the

operational stage they help in structuring the generation of a model as described in Figure 4.4. This conceptual-to-implementation workflow is a desirable feature in a Industry 4.0-oriented EM framework, as the most recent EM frameworks (e.g. RAMI 4.0) seem to lack it.

4.3.3.2 Genericity Dimension

As with CIMOSA, the genericity degree is about the reusability of modules. The genericity dimension of MEMO I4.0 allows adapting the modeling level in function of the context, and provides a maximum agility for reusing already developed models. We define three degrees of genericity of modules:

- *Generic* that can be used in all contexts;
- *Partial* that could cover more than one context, but not all;
- *Specific* which is adapted only to a specific context.

4.3.3.3 Scale Dimension

The scale dimension allows for tuning the overall model granularity and facilitate model evolutions. This dimension can be defined (as in RAMI 4.0) as the modeling granularity level of modules. The same production process can be, e.g., described at the work center, line, workstation, or even product level. Modules that differ in scale can be created for the same activity and genericity degree.

4.3.3.4 View Dimension

The view dimension of MEMO I4.0 has the same purpose as in the existing EM frameworks, i.e., it simplifies the modeling task and supports generating a compatible model regarding needs, objectives, and system features. Hence, it acts as a support for respecting the interoperability principle during the development of each module and contributes to different phases of the strategical/tactical stage to ensure global consistency. Therefore, MEMO I4.0 introduces the view dimension to manage the modeling complexity.

As we presented before, the view dimension allows us to ensure this consistency, but which are the required views that cover the modeling of the smart factory in the Industry 4.0 environment? As specified by Noran (2004), the selection of views for modeling depends on the scope and purpose of the modeling process. In the last chapters, we introduced four conceptual models of view: the function, the decision, the information, and the performance. In this chapter, we show that we need other views for comprehensive enterprise modeling.

The process view and the resource & organization view are important in classical EM frameworks. The process view defines plant configurations and provides details of products and processes, while the resource & organization view supports the definition of production resources, their capabilities, and their assignment to workstations as reported by the well-known PPR model (see Section 3.4). Therefore, we decide to add these two views in our modeling and define their conceptual models. In Industry 4.0, the integration of data from smart devices is a major challenge, as the data needs to be transformed into information. Adding a data view is necessary to ensure connectivity between cyber and physical systems. Figures C.1 to C.4 from the Appendix C illustrate the conceptual models of, respectively, the partial integration of the function, decision, performance and information views, as well as the process, data and resource & organisation views of an MS. Integrating the latter views with the conceptual model of the partial integration of the former ones ensures consistency from the business layer to the asset layer.

Table 4.2 summarizes the dimensions of MEMO I4.0 and how they respond to its key principles and modeling objectives.

Table 4.2: Correspondance between MEMO I4.0 principles, objectives and dimensions.

EM principles	**Modeling objectives**	**EM dimensions**
Modularity	Structure the modeling process	Life cycle
Agility	Adapt the modeling levels	Genericity
Robustness	Tune modeling granularity	Scale
Interoperability	Ensure the modeling consistency	View

MEMO I4.0 dimensions need to be developed according to their corresponding key principles. Figures C.6 to C.9 from the Appendix C illustrate respectively the relationships between the genericity, life cycle, view, scale dimensions and the agility, modularity, interoperability, and robustness principles in order to fit Industry 4.0 modeling requirements.

Figure 4.6 provides the conceptual model that describes the relationships among principles, objectives and dimensions.

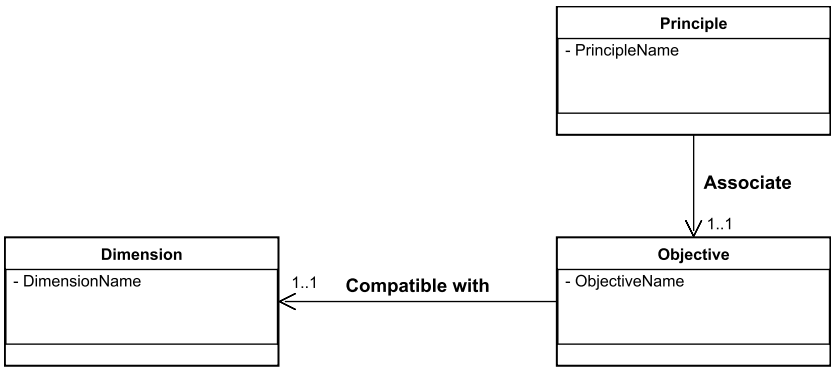

Figure 4.6: Compatibility between modeling objectives and dimensions of EM framework.

4.3.4 Specifications and Purpose of MEMO I4.0

So far, we have presented the key principles, modeling objectives, and framework of MEMO I4.0, and used UML diagrams to illustrate the consistency among them. Figure 4.7 integrates previous conceptual models in one model. We use red color only to highlight the links added due to the context of Industry 4.0.

In order for MEMO I4.0 to offer industrial companies and the research community an Industry 4.0-oriented methodology capable of modeling enterprise systems and build DSSs for different use cases, the idea is to have a library of reusable modules, each one describing one process of an enterprise activity. We propose to illustrate our purpose using the UML diagram. Figure 4.8 shows that on the one hand, the library is composed from several modules, which are composed from several submodules. On the other hand, the module is obtained from one or several models, which are the instance of metamodels.

Of course, MEMO I4.0 must take the requirements depicted in Figure 4.7 into account in its definition, as well as specify the definition of the library of modules and the building of the DSS. Figure 4.9 gives the starting point for developping MEMO I4.0. Once more, the red color points out the links imposed by the context of Industry 4.0, while blue color is used to show the ultimate goal of MEMO I4.0.

In terms of modeling, we could summarize specifications imposed by key principles and dimensions of MEMO I4.0 as follows :

- Develop data-driven models;
- Adopt an incremental development of models (module by module) and separate different flows;
- Use a mapping layer between model and specific development platform;
- Choose the same granularity for the submodules that must be integrated into the same model.

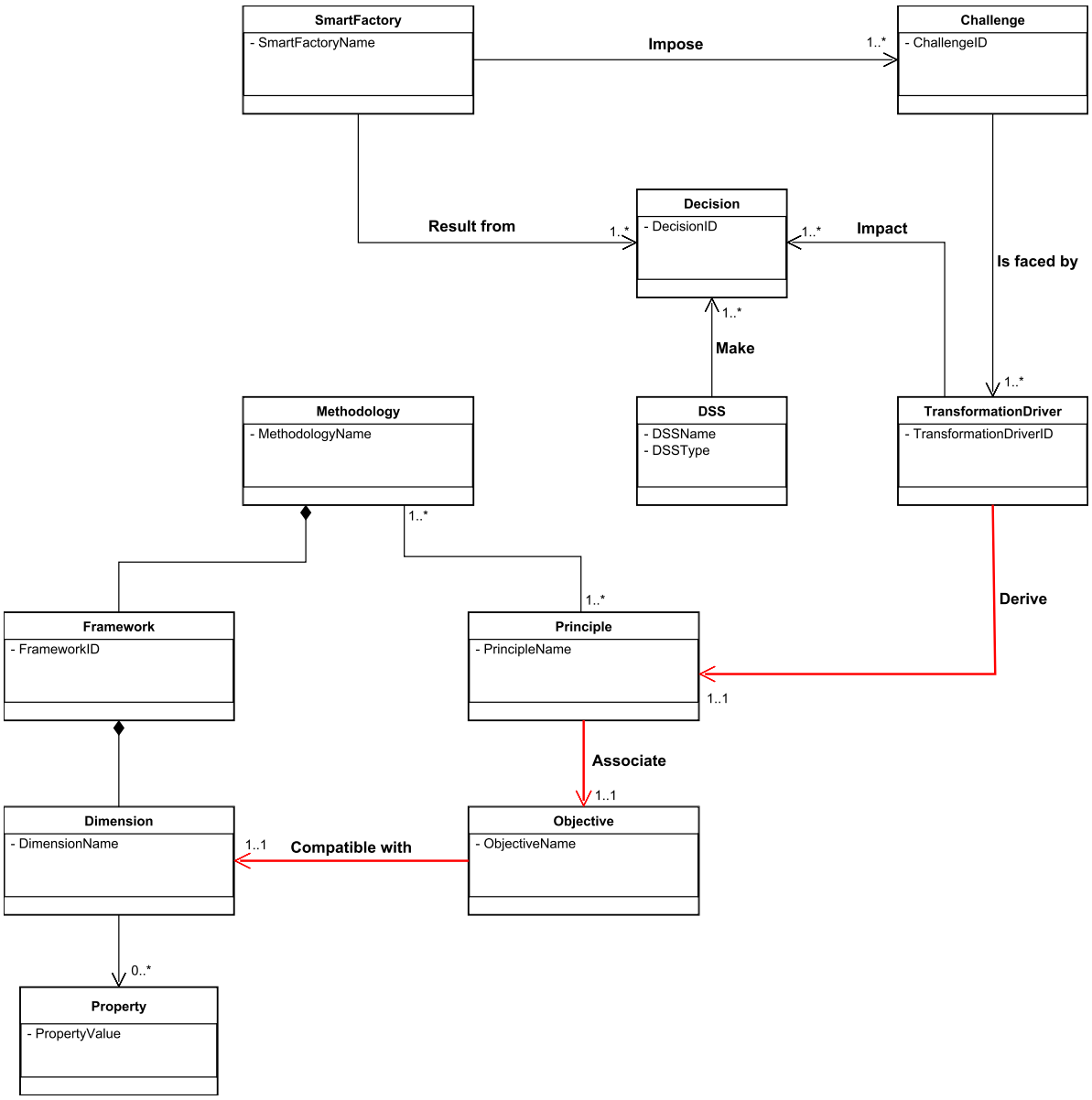

Figure 4.7: Conceptual model illustrating relationships among challenges and their main transformation drivers in Industry 4.0, and the principles, objectives and dimensions of MEMO I4.0.

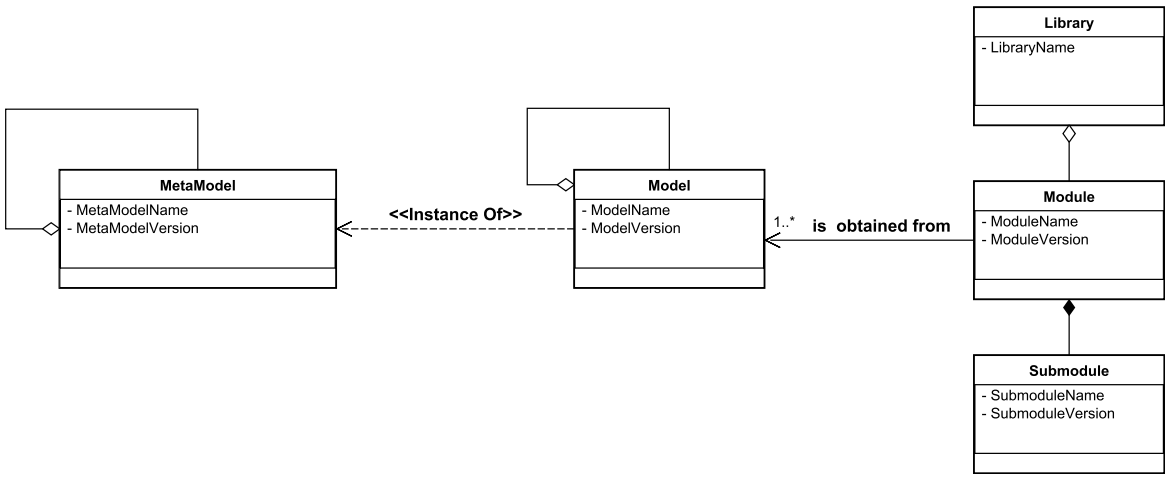

Figure 4.8: Conceptual model illustrating the concept of MEMO I4.0 library.

Developing models based on these key principles leads to better horizontal & vertical integration.

4.4 Application of MEMO I4.0: Strategical/Tactical Stage

After designing MEMO I4.0, we propose the following structured approach to implement it in an enterprise environment, starting from the strategic/tactical stage.

Figure 4.10 proposes a roadmap defining the steps to follow in the strategical/tactical stage of the development of modules. The Figure provides only the main steps in each phase of the life cycle: a complete version of the roadmap is given in Figure E.1 with more details for each dimension.

In Figure 4.10, the three columns illustrate the dimensions: Scale, Genericity, and View. The phases of the life cycle dimension, namely, definition, design, implementation, and maintenance are represented by the four rows.

During the definition phase, the primary objective is to establish the requirements and specifications of the modeling process. This involves determining the scale level, the degree of genericity, and the specification of elements that build up in each module.

In the design phase, the focus shifts towards creating the model of each module and the integrated model resulting from several modules. To accomplish this, appropriate modeling choices are made according to the desired scale level and the modeling approach that effectively achieves the desired genericity degree. Additionally, any events impacting decisions, information, or performance are identified, and metamodels are instantiated to fulfill the requirements specified in the definition phase.

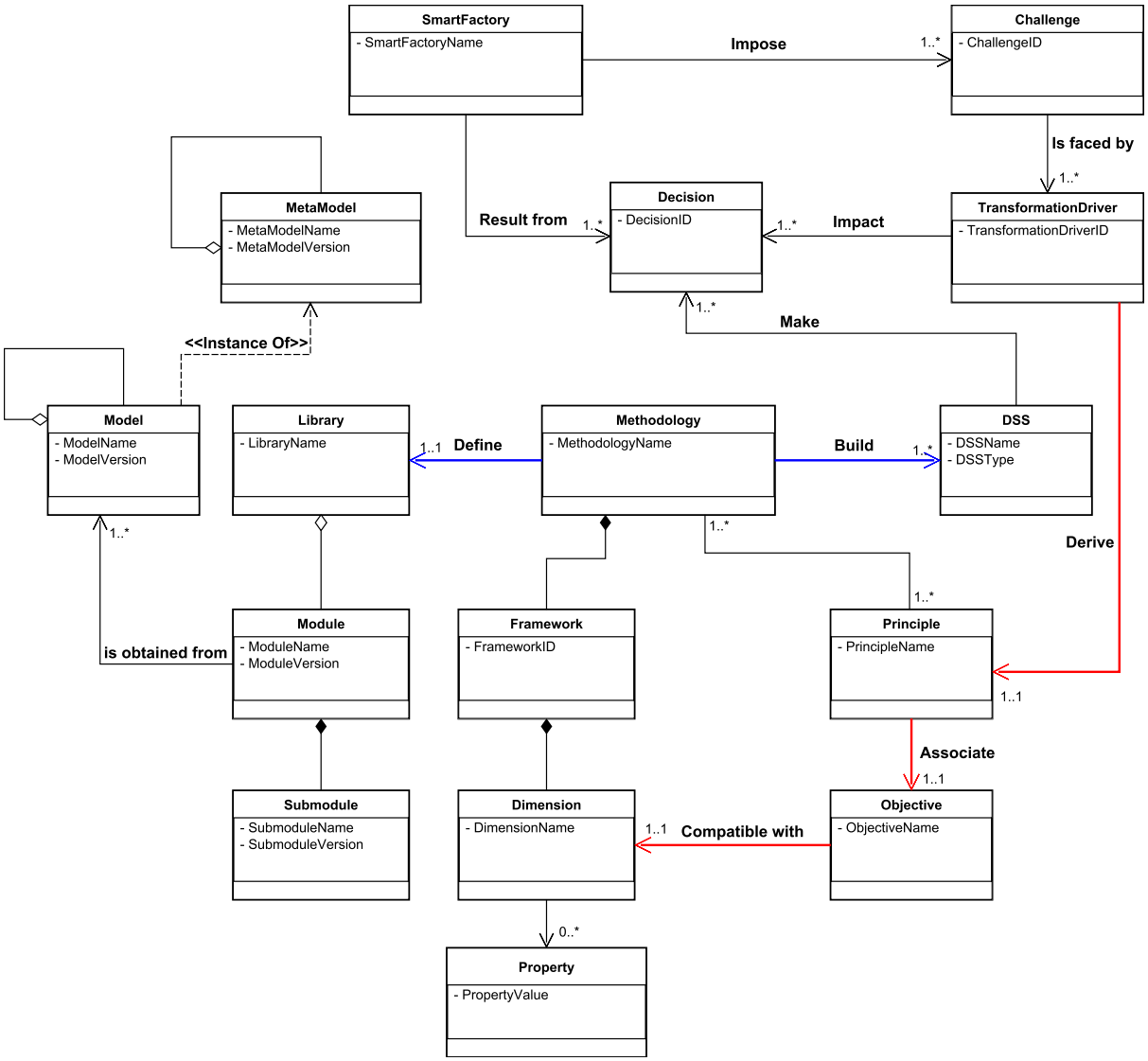

Figure 4.9: Conceptual model illustrating the MEMO I4.0 purpose and specifications.

Next, the implementation phase aims to verify and validate the behavior of each module and the integrated model of modules. Initially, pseudo-code is generated to describe the business logic of each module, and a four-tier architecture layer encompassing the data access layer, business layer, presentation layer, and mapping layer is created. Subsequently, the simulation-optimization model is built, and experiments are conducted to assess the expected flexibility and agility.

Finally, in the maintenance phase, the focus is on the versioning of the model of each module, generating relevant documentation, and storing the version in the module library. Prior to this, a review of the scale level and genericity degree is conducted, and compatibility among views is checked. Based on the outcomes of a review meeting, a decision is made to validate or reject the module version.

In what follows, a fictitious enterprise, named E1, will be used as a didactic example that will serve to illustrate the steps of the MEMO I4.0 deployment. The primary process of E1 is assembling parts to build vehicle motors, but the example could as well apply to enterprises of other types and purposes.

4.4.1 Definition Phase: Requirements and Specifications of Modeling

A strategy to build a digital twin is defined that takes into account the company DNA, guidelines and the vision towards Industry 4.0 as the context in which the change of decision-making process, the evolution of the types of MS, and the emergence of new KPIs (El Alaoui El Abdellaoui et al., 2019) will take place.

In this phase, SADT (Structured Analysis and Design Technics) is used to yield a systemic view of a complex system based on a hierarchical structure, in which each activity is explained in detail in a process workflow diagram. This task can be achieved by using interviews or Gemba walk if the MS already exists. Likewise, we identify the functional specifications using reference documents and interviews with the activity managers. The objective is to understand needs, objectives, most frequent problems, and decision workflow.

Further, KPIs are defined in detail for each module. In order to simplify the maintenance, the information is grouped, and the resource set is identified, as well as the related roles and data, according to the department in charge of it. These elements require choosing the most-fitting scale level since they will impact the modeling granularity in the design phase.

4.4.1.1 Definition of Modeling Objectives and Scope

Usually, brainstorming sessions are organized to discuss the objectives that the managers seek to achieve, in order to delimit the scope of the modeling.

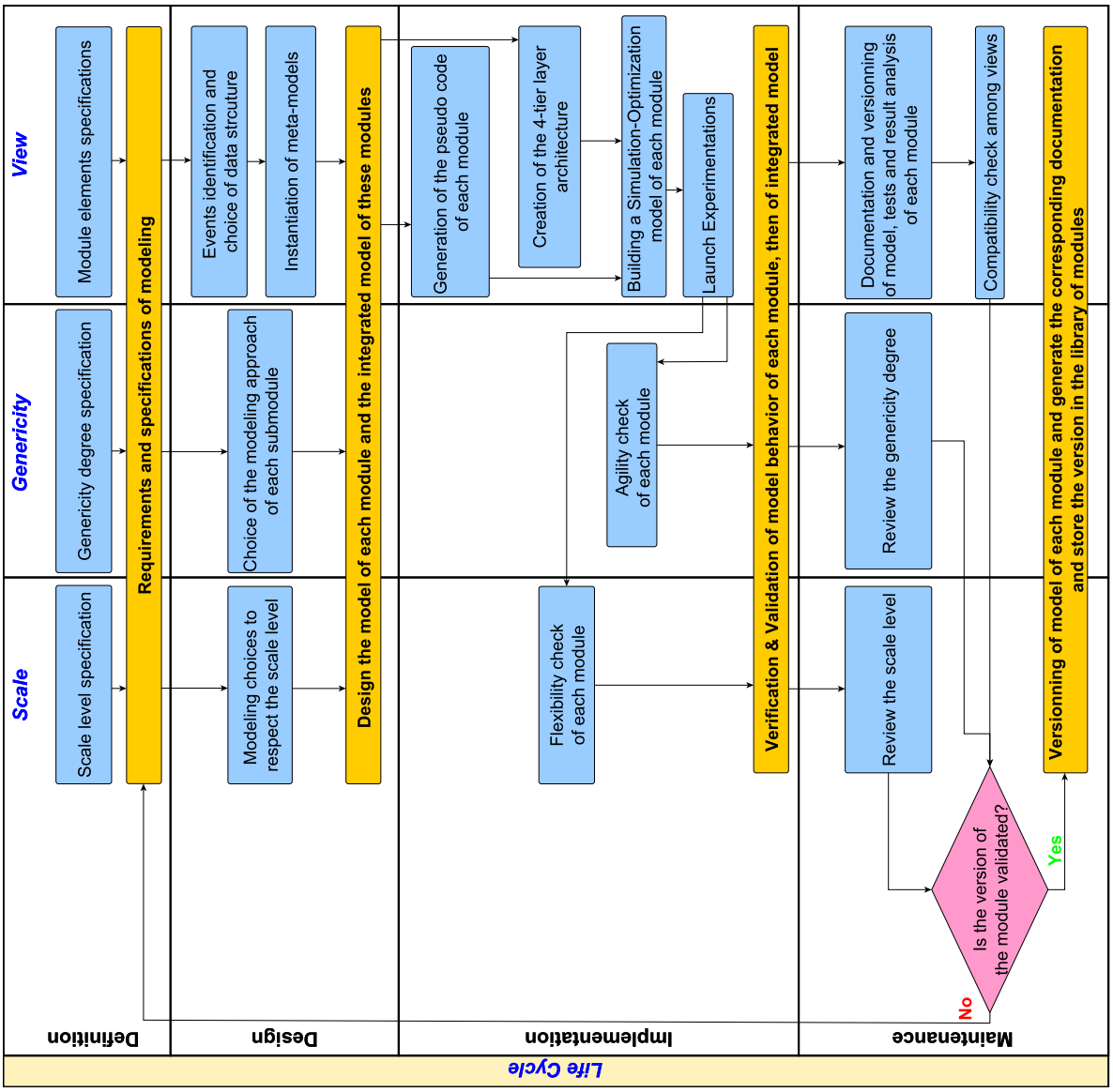

Figure 4.10: Roadmap for implementing MEMO I4.0 (compacted version).

The main objectives of enterprise E1 are:

1. increasing the utilization rate of the production resources.
2. reducing the cost of the production.
3. improving the production schedule.

According to these objectives, we delimit the scope of the case study and MEMO I4.0 fictitious deployment to the production activity.

4.4.1.2 Description of the Enterprise Vision, Guidelines & DNA

Enterprises often use the SWOT (Strength, Weakness, Opportunities, and Threats) analysis to evaluate the competitive position of the company. It is a commonly used tool for strategic planning (Phadermrod et al., 2019). The SWOT tool allows to determine the internal and external factors that can have an impact on the business. Furthermore, it helps to summarize the current situation and foresee potential future directions for the business.

In the E1 case study, the result of the SWOT analysis is shown in Figure 4.11.

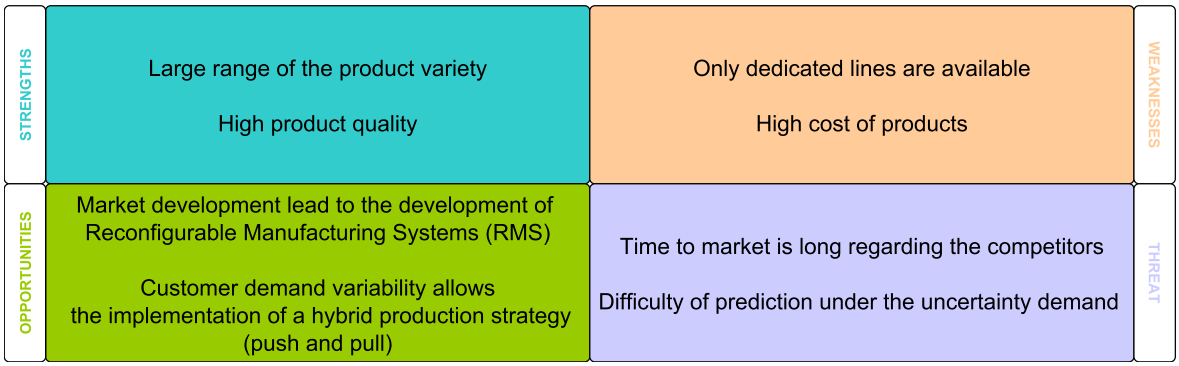

Figure 4.11: The SWOT analysis to guide the enterprise vision E1.

The enterprise DNA is often consolidated within the enterprise handbook, while valuable guidelines are derived from the point of view of experts and insights gained through Gemba walks. In the definition phase, capturing the DNA and guidelines of the activities involved in the modeling scope is crucial as they profoundly impact the modeling choices during the design phase.

DNA and guidelines for the production activity of E1 include employing the job shop as a production routing strategy, implementing the push policy for producing a variety of products, and organizing the production schedule into two shifts per day.

4.4.1.3 Decomposition into Modules and Definition of their Functional Description and Assumptions

SADT is a method used to yield a systemic view of a complex system based on a hierarchical structure. It uses interviews or Gemba walk if the MS already exists to explain in detail each activity in a process workflow diagram.

As depicted by the function view (see Figure 3.2), each activity comprises one or several functions. These latter provide a functional description of the system at study, i.e., the business layer description primarily formalized through the guidelines. In this step, we define a dedicated module for each function of the plant activity at study, and we use the enterprise vision, the real system when it exists, and the modeling simplification to formalise the assumptions. The same notation previously defined for the MS features is used to express these assumptions.

In the case of E1, we decide to focus on the production activity, which is composed of three main functions, namely *Plan*, *Schedule* and *Assembly*: for each of them, we will need to define an appropriate module. Figure 4.12 illustrates an SADT diagram that involves the functions based on customer requirements and demand forecasting. The diagram considers inventory, resource capacity, and bill of materials (BOM) constraints first (up, left), the generation of the planned manufacturing order or purchase order. Next, the diagram shows (center) the schedule function, i.e., how the planned manufacturing order is transformed into a manufacturing order. According to the manufacturing order description, the assembly function (bottom, right) is responsible for producing blocks/products.

In Appendix E.2, Table E.1 provides the list of assumptions identified for the E1 modules.

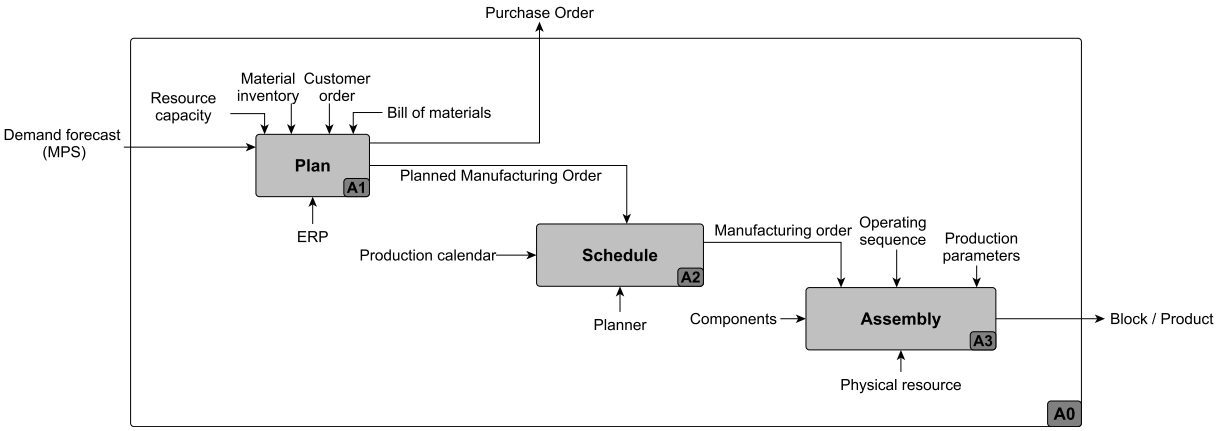

Figure 4.12: Example of SADT diagram for E1.

4.4.1.4 Separation of Flows and Definition of Submodules

The modularity principle of MEMO I4.0 recommends using an incremental development (module by module) according to the life cycle dimension. Based on the workflow described in the SADT diagram, the process view focuses on the separation of the flows (physical, information, and control) for each module. This latter is subdivided into submodules compatible with the associated flows, hence a physical, an information and a control submodule. This decomposition provides high flexibility since it allows to change the control part without changing the physical part or to use the same control part for different structures.

Table 4.3 provides the separation of flows of the three modules involved in the production activity of E1, and Figure 4.13 illustrates the exchange of flows among the three submodules of the assembly module.

Table 4.3: Flow separation for the enterprise E1 modules.

Flow type	Plan	Schedule	Assembly
Physical	-	-	Blocks/Products
Informational	Customer Order Purchase Order	Planned manufacturing order	Manufacturing order
Control	Demand control	Shift control	Production control



Figure 4.13: Illustration of the modularity principle via an example of decoupling flows (physical, information, and control) of the assembly module of E1.

4.4.1.5 Building Modeling Scenarios

In this step, the aim is to review the concepts related to the modeling of the modules, and to establish scenarios that will lead to choose the most appropriate genericity degree.

The first option to consider is whether examples that are similar to the module at study exist in the literature, due to e.g., previous implementation in case studies related to industrial companies. The second point is to use the existing enterprise guidelines and best practices to include in the modeling. The last point is to integrate the enterprise vision, based on the market analysis and competitor positions.Also, it is recommanded to use a matrix to establish several potential scenarios of current (As-Is) and future (To-Be) situations.

For simplification reasons, we provide only details about the Assembly module of E1 (concepts related to software part, system structure, product Variety, and others have been reviewed in Subsection 1.2.1.1). Although three types of MSs exist (DMS, FMS, RMS), based on the enterprise guidelines, the production of mix products deprecates using DMSs. Moreover, according to the E1 vision, we expect to develop a RMS in the future.

Six scenarios seem therefore worth investigating, as shown in Table 4.4. Scenarios 1, 4 and 5 do not consider any change in the current MS: in terms of modeling, the model can therefore represent only the As-Is situation. Scenarios 2,3, and 5 will consist of modeling the To-Be situation.

Table 4.4: Potential scenarios of transformation of the MS type to consider when modeling the assembly module of E1.

Current MS \Future MS	DMS	FMS	RMS
DMS	1	2	3
FMS	X	4	5
RMS	X	X	6

4.4.1.6 Specification of the Modeling Context and the Genericity Degree

According to the enterprise vision, the genericity degree is decided so that the module to develop integrates both current needs and possible future evolutions.

The model capable of representing the As-Is and To-Be situations should be able to encompass them all and hence be more generic of them. The ability to model different scenarios represents the degree of genericity of module's model, as the reusability of the model in a different modeling context depends on it, as shown in Table 4.5.

Table 4.5: Genericity degree in function of modeling context.

Genericity degree	**Modeling context**
Specific	As-Is situation
Partial	As-Is situation and at least one to-be situation
Generic	All scenarios (As-Is and To-Be)

In the case of E1, we understand from the SWOT analysis that the enterprise has the ambition to increase the product variety and seeks high flexibility to reconfigure its production rapidly. Therefore, we fix the genericity degree at the generic level for the Assembly module.

4.4.1.7 Identification of Decisions related to Modules

The work already done on Chapter 2 allows us to choose for each module the decisions that managers and directors consider to be the most-fitting. Note that these decisions must be consistent with the objectives already defined in Subsection 4.4.1.1.

Regarding the objectives expressed by the managers of E1, Table 4.6 defines decisions to be considered for each module of the production activity.

Table 4.6: Decision linked to the modules of the production activity of E1.

Plan	Schedule	Assembly
Demand forecasting	Sequencing & Scheduling	Lot sizing & Scheduling

4.4.1.8 Definition of Module KPIs

With the team project, for each module, we define KPIs able to assess the performance of the function, but also that are consistent with the objectives fixed.

For E1, Table 4.7 lists the KPIs chosen for the three modules.

Table 4.7: KPI of each function of the production activity of E1.

Plan	Schedule	Assembly
Monthly average of demand per product	Daily average of manufacturing orders	Overall equipment effectiveness

The per-month average demand per product is considered as an appropriate KPI to evaluate during simulations to help improve the demand forecasting performed by the plan function. The daily average of the manufacturing orders is considered capable to improve the sequencing and scheduling decisions, due to its impact on the production schedule. Finally, the overall equipment effectiveness (OEE) can help tracking the utilization rate of resources.

4.4.1.9 Specification of the Scale Level

After the definition of the KPIs of each module, we must choose its scale level, i.e., the modeling granularity. The scale level impacts the flexibility of the model. It is essential to adapt it based on the analysis we want to conduct using the model, as the time spent developing a detailed model is not a good investment if such details are not necessary for understanding the module behavior.

The first aspect to define is to delimit the study context and specify what the model is supposed to explain. The second aspect is to determine which are the parameters that impact the behavior of the module. The best compromise must be found, between a large

number of parameters, which allow for a finer granularity, and a lower number, which allow for simpler data structures to be handled. Another element impacting the choice of the scale level is the decisions that the module is supposed to address. CIAOC features are required to make these decisions and to adjust the modeling granularity.

When it is possible, it is recommended to have the same scale level for all submodules of the module to avoid flexibility problem. A good practice is usually to think in terms of the information submodule since it is the interface between the physical and control submodules. Furthermore, the scale level of a module is usually determined by the scale of the information submodule.

Appendix E.2 provides an example of choosing the scale level of the Assembly module. In addition, the information submodule (Management of Manufacturing Order) is used to illustrate how the choice of the scale level must be made. We demonstrate that the scale level can be fixed at the work center or line level if the manufacturing order is managed at the ERP level, mainly due to the cost aggregation requirement. However, it is preferable to fix the scale level at the line or workstation level when the manufacturing order is managed at the MES level since the resource configuration must be done accurately. Finally, the scale level is fixed at product level, when a real-time tracking of manufacturing order with a precision of produced serial numbers is required.

4.4.1.10 Definition of Departments linked to Modules

This step aims to identify the departments involved in the (sub)modules and the organizational units of each department that contribute to defining and updating them.

Table 4.8 provides an example of departments involved in the production activity of E1, as well as organizational units linked to each module.

Table 4.8: Departments involved in each module of the production activity of E1.

Function	*Plan*	*Schedule*	*Assembly*
Department	Manufacturing Logistic		
Organizational Unit	Forecasting Engineering	Scheduling	Production Maintenance Continuous Improvement

4.4.1.11 Definition and Association of Information of Modules with MS feature

This step aims to define information required for each module by refering to the involved MS features. The information view presents a snapshot of an MS in real-time, which includes the values of all MS features, and any entity in the studied MS either produces or consumes the value of a specific feature. Sometimes, it is possible a piece of information

serves as a setting for specific business processes or can be used for the tuning of simulation and optimization model: therefore, it can not be considered as an MS feature.

The partial integrated model of views (see Appendix C.1.1) shows the relationships of the information view with other views, particularly, decision, performance, and function. We take into account this relationship to simplify the definition of the information related to each module. We use decisions specified in Subsection 4.4.1.7 to obtain the CIAOC properties required, hence the involved MS features. Then, based on the KPIs defined in Subsection 4.4.1.8 we deduce information related to them. Finally, we exploit functions defined in the Subsection 4.4.1.3 to detect information consumed and produced. The concept of function data serves to group data related to one module (see Subsection 4.4.1.12).

Table 4.9 shows the required features for the E1 case and their unique code. For the sake of simplicity, we consider in this table the CIAOC properties as function data.

Table 4.9: Example of information related to the schedule module of E1.

Module	Function Data	Code	MS Feature Name
Schedule	**Constraint**	P.p.15	Sequence of operations assigned on same Machine
		P.p.19	Buffer Size
		P.c.1	Constraint of processing operations in Machine
		P.c.6	Precedence constraints between tasks
		R.c.15	Capable Machine per operation
		T.p.14	Completion time
		T.c.11	Setup-time of Machine
	Inputs	P.p.9	Number of operations per job
		Z.g.3	Number of jobs
		R.g.2	Number of Machines
		T.g.1	Set of periods
		T.g.2	Number of periods
		T.p.1	Processing time of job/Machine
	Output	P.p.16	Production sequence
		T.p.10	Starting and completion time of product per Machine in period (t)
	Criteria	Z.o.1	Minimize the number of tardy jobs
		T.o.3	Minimize makespan
		T.o.4	Minimize the flow time
		T.o.5	Minimize the maximum tardiness
		T.o.6	Minimize Mean Square Deviation of completion times

4.4.1.12 Identification of the Raw Data and per-Department Organization

Usually, raw data can be divided into four categories:

- Transactional data, describing business events that occur most of the time in enterprise information systems, e.g., reception of new customer orders.
- Master data, which are necessary along transactional data for the business process, e.g., the customers.

- Reference data, which help in interpreting data records and are shared by enterprise information systems, e.g., customer order status.

- Metadata, providing information about other data, e.g., data source.

However, in MEMO I4.0, regardless of the data category, we refer to data by what we call *data set collection*, and we specify their *data source*, and which *department* is responsible for their management. As explained in the conceptual model of the data view, either enterprise information systems (EIS) or messages provide the data set collection. However, we think that data set collections must be grouped based on their use in the modeling, i.e., according to the module requiring them. In MEMO I4.0, this group is called *function data*, and we generate it using data set collections extracted from EIS or messages either by using them as they are or after some transformations such as processing, analysis, and aggregation.

4.4.1.13 Identification of Elements Involved in Each Module

This step is critical in the definition of the module since it impacts the modeling choices made in the next phase. The goal is to define elements involved in the module, i.e., key concepts that must be considered in the modeling process to formalize the module behavior. SADT diagram, information and data required by each module are used to identify module elements.

For instance, Table 4.10 provides the list of elements involved in each module of E1.

Table 4.10: Elements of E1 included into each module.

Elements	**Plan**	**Schedule**	**Assembly**
Workcenter	x	x	x
Cell		x	x
Workstation			x
Product	x	x	x
BOM	x	x	x
Component	x	x	x
Operation			x
Process	x	x	x
Shift		x	x
Period	x	x	x
Demand	x		

4.4.2 Design Phase: Instantiating Metamodels of Views

In this phase, the scale level decided in the definition phase allows adjusting the modeling granularity and, ultimately, the flexibility of modules. We make the modeling choices according to the chosen genericity degree. Modeling choices are influenced by the guidelines but potential evolutions that may be interesting to investigate can be included.

Based on the SADT diagram, we specify the events that trigger each function, and according to the defined decisions, we identify the events that may impact the information used by the models.

Besides, the computation of the already identified KPIs (see Subsection 4.4.1.8) and their structure is defined, and the required information are detailed. Tree structures are preferred in case many-to-many relationship exist between the entities, otherwise, a 2D/3D matrices are sufficient.

We instantiate the metamodel diagram of each view to obtain the models of a specific module or the integrated model of modules that are compatible with the SADT diagram and include all information related to the decision process.

4.4.2.1 Decision of the Modeling Choices of Each Submodule

The modeling choice of each submodule of a module concerns how to model the elements involved in this submodule (see Subsection 4.4.1.13). The scale level at which each element must be managed depends on the business process, and impacts the flexibility of module, i.e., effort to make for switching from one scenario to another. The classical modeling choices are : entity, attributes of entity, queue, delay, resource, process, set, and advanced set. Definition of these modeling choices is given in Table G.1, and Table E.2 provides details about the modeling choice for each submodules of the E1 modules. We provide also the period control submodule, which is shared among all modules.

4.4.2.2 Choice of the Modeling Approach of Each Submodule

In Section 4.2, we defined the three approaches that could be used for modeling when dealing with the genericity degree of module. In this step, we adapt the appropriate approach for each submodule (physical, control, and information) of the module with the conformance of the generic degree fixed at the definition phase. Note that different scale levels (modeling granularity) can be chosen for the same genericity degree (see Figure 4.14).

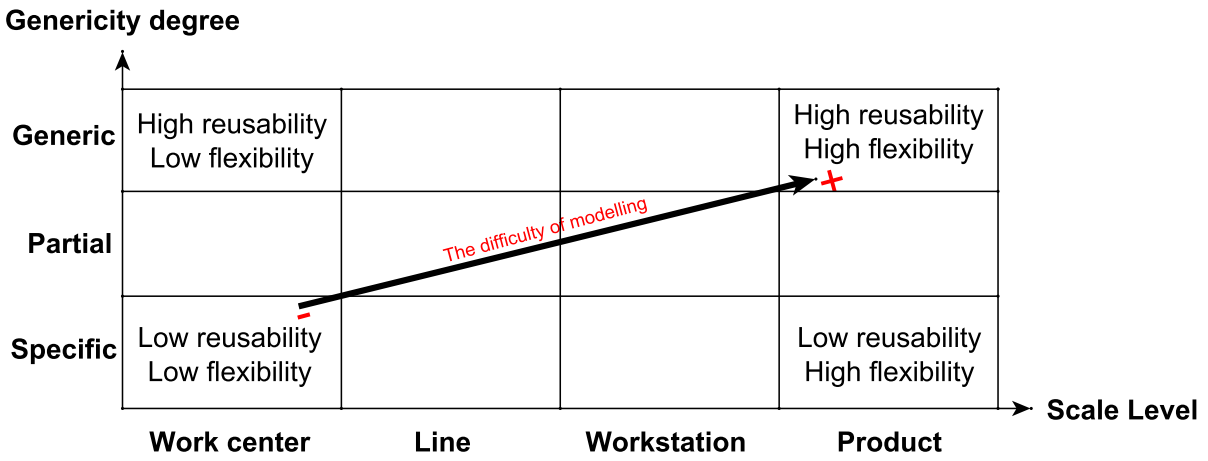

Figure 4.14: Reusability and flexibility of modules based on the genericity and the scale dimensions.

In Appendix E.2, we propose to give details only about the physical submodule (MS structure) of the Assembly module of E1. The genericity degree for this submodule has been fixed at a generic degree at the definition phase, which means that we must be able to model the DMS, FMS, and RMS using the submodule. Based on the example in Appendix E (Figure E.3), we investigate using concepts such as station, cell, and stage for structuring production activity in DMS, FMS, and RMS, respectively. We highlight the importance of choosing an appropriate approach (independent, intermediate, or integrated) during modeling, considering the complexity, and accurately analyzing common features and differences.

4.4.2.3 Specification of Function-Triggering Events

The conceptual model of integrated views (see Figure C.5) illustrates that each function (module) can be triggered by one or many events. This step focuses on identifying these events. Its aim is to reinforce the interoperability among modules by ensuring their decoupling, since, each module is triggered based on events independently of behavior of other modules. The SADT diagram helps to understand how each module of the process is triggered. Moreover, separating different flows makes identifying events easy because the information flow necessarily illustrates these events.

Table 4.11 provides the events that trigger the modules of E1.

4.4.2.4 Decisions Impacted by Update Events

The objective is to specify the events that will impact decisions already identified in the definition phase. In other words, events will lead to an update on the decision. Often, we

Table 4.11: The events that trigger the modules of E1.

Function(module)	**Event**
Plan	Receive of the customer demand Reach the planning period
Schedule	Release of planned manufacturing orders
Produce	Schedule of manufacturing orders

resort to information related to the module and select those included either on the inputs or constraints of the decision.

For instance, in the case of E1, the scheduled status of the manufacturing order is the event that triggers the function *schedule* and acts on the decision of *production scheduling*.

4.4.2.5 Specification of the Events Impacting on Each Information

The conceptual model of the information view (see Figure 3.5) illustrates that an event occured can impact on one or several MS feature. This step aims to map the events triggering each function identified in Subsection 4.4.2.3 with the impacted MS feature. The goal is to ensure that any event occured in the enterprise is correctly taken into account and coordinated with business processes in standardized manner through using MS feature.

In the case of E1, remark that an event such as the reception of the customer order will have an impact on the feature *number of orders per day*. Likewise, the release of the manufacturing order event will impact on *the quantity of product to be produced*.

4.4.2.6 Building the Structure and Format of KPIs

For each KPI identified by managers during the definition phase, we specify the KPI structure. This includes, the formula, the unit of measure, and the frequency of measure. Also, we determine the KPI representation, i.e., the way of displaying the KPI value (e.g., table, plot, bar graph).

Table 4.12 gives details about computing KPIs identified for the modules of the production activity of E1. Subsection 4.4.3.15 shows an example of representation of OEE KPI in form of line plots, where X-axis represents (Period), and the Y-axis, OEE value.

4.4.2.7 Data Structures

The data related to each module serves as the functional data, encompassing information essential for representing the business logic inherent in the module. These data related to one module are grouped and called function data.

Table 4.12: Example of KPIs to evaluate for the production activity of E1.

Name	Formula	Unit of measure	Frequency of measure	Value
Monthly average demand per product	$\frac{Number\ of\ orders\ per\ day \times Number\ of\ periods}{Number\ of\ products}$	[#]	Monthly	1500
Daily average manufacturing orders	$\lceil \frac{Produced\ quantity}{Lot\ size} \rceil$	[#]	Daily	12
Availability	$\frac{Number\ of\ shift \times Machine\ availability\ time/shift}{Daily\ working\ time}$	[%]	Daily	92,5%
Performance	$\frac{Produced\ quantity}{Number\ of\ product\ to\ be\ produced}$	[%]	Daily	95%
Quality	$Product\ quality \times 100$	[%]	Daily	97,0%
Overall Equipment Effectiveness	$Availability \times Perfomance \times Quality$	[%]	Daily	85,23%

As defined in the conceptual model of the data view,the function data related to each module are generated by the data set collection, which is provided by the enterprise information systems or builded from smart asset messages. For each data set collection, we need to specify the data structure that will be used. In addition, based on elements defined in Subsection 4.4.1.13, we define a unique ID of the data set collection.

Table 4.13 provides an example of data structure of data set collection involved in E1 modules. ";" indicates the separation between each dimension.

Table 4.13: Data structure of data set of the E1 modules.

Data set collection	Data structure	Unique ID
Demand	1D	(Product, Period)
BOM	Tree structure	Workcenter, Cell, Workstation, Product, Component
Inventory	1D	(Product, Period)
ResourceFeatures	3D	Period; Shift; (Workcenter, Cell, Workstation)
ProductionSequencing	2D	Period; rank
ProductionScheduling	2D	Period; Product
ProductionCalendar	3D	(Workcenter, Cell); Product; Period
ProductionParameters	2D	(Workcenter, Cell, Workstation); Product

4.4.2.8 Drawing an Organogram

An organogram allows to have an organization view, i.e., depicts the existing departments, their organizational units and their roles. The first reason for using an organogram in MEMO I4.0 is to obtain a list of collaborators that are or could be involved in the project. Another reason is to enumerate available skills in the company organization, which could be used during the life cycle of the MEMO I4.0. Finally, the ultimate reason is to identify responsible actors for data management.

4.4.2.9 Instantiation of the Metamodels of Views

It is the core step in the design phase of the development cycle, because, all details defined before must be taken into account. Based on already defined metamodels of MEMO I4.0 views (C.f. Figures 3.2, 3.4 to 3.6 and C.2 to C.4), now we instantiate the metamodel of each view to capture the business logic, guidelines, DNA, and constraints of the company related to each module.

We use the Object Constraint Language (OCL) to define the rules that must be applied to the UML diagrams already defined. In addition, all generalization used must specialized for the company at study, as well as the multiplicity of relationships.

Figure E.4 provides an example of the instantiation of one metamodel of the function view for the modules of E1. Note that we present two equivalent ways to instantiate the metamodel, either module by module, or by considering all modules in the same model.

4.4.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module

In this phase, once the technical solution to capture the data raw is chosen, the next step is to create tables related to the module in a database (DB) based on the data model to ensure data integrity and compatibility with the metamodel diagram.

With the database tables in place, the next step is to build the business model layer. This layer incorporates insights intrinsic to the system at study and is independent of the implementation platform. Then, the mapping layer must be created to establish connections between the business model layer and the model implementation in a specific platform. After that, a Pseudo-code of the module is written to facilitate the implementation, providing a high-level, human-readable representation of the logic required.

Depending on the DSS type targeted, the development of an optimization or a simulation, or a simulation-optimization-based model follows. Afterward, the appropriate solving approach for the optimization model is selected, and numerical experiments are conducted to check that the solving approach is adapted to deal with real instances. Then, the simulation model is built and used for testing modules in a virtual environment before actual implementation. It is crucial to accurately map the decision variables from the optimization model to the corresponding attributes in the simulation model. This mapping ensures that the results obtained from the optimization model can be precisely applied in the simulation.

Traceability files for each module, including its attributes and variables, are created to ease the model verification. Integration of modules is carried out, and the expected behavior is verified. Further, corner tests are run to ensure the stability of the model.

The KPIs of each module are displayed and analyzed to evaluate the results of the model against the predefined objectives. Finally, experimentations are launched first to check for each module whether the genericity degree fixed is respected, then to check if the flexibility is achieved w.r.t. the chosen scale level.

4.4.3.1 Choice of the Technical Solution to Capture the Raw Data

This step allows choosing the technical solution to capture the data raw. Often, performing data processing is needed to obtain an information from raw data in forms of MS feature by using the data access layer (see Subsection 4.4.3.3).

Table 4.14 provides an example of technical solutions selected to capture the raw data, and shows the information extracted from these data.

Table 4.14: Technical solutions to capture raw data.

Raw data	Technical solution	Information
Total time of operations of MO Quantities consumed in operations of MO	ERP	Production cost of product in period (t)
Time of leaving the workstation Time of arriving to the next workstation	RFID tag	Travel time between workstations
Time of reception of demand Lead time required Time of producing the demanded quantity	PMS	Service level
Start time of operation End time of operation	Sensors	Processing time

4.4.3.2 Creation of the Database Tables Required by the Data Model

After designing the conceptual models corresponding to the module, we transform the classes defined in each view into tables taking into account the OCL constraints. This transformation respects the classical rules adapted for obtaining the physical data model from conceptual data model (e.g., the transformation of the many-to-many relationships with adding a new table with a primary key referencing the two keys of the two tables).

However, before creating such tables, we need to make decision about the database management system. The choice depends on the enterprise information systems already existing in the company, and the effort required for the integration of such database management system.

The database is responsible to store the MS features of the company at study, as well as to store the historical values of KPIs and maintain them available for those who need access.

4.4.3.3 Creation of the Data Access Layer

The data access layer plays the role of transforming information into the correct format adapted to the model of the enterprise. Usually, the data access layer consumes information from JSON (JavaScript Object Notation) files generated by the EIS or rarely directly from the database of these systems. The ultimate goal of the data access layer is to create a Data Access Object (DAO), which is responsible for interacting with the database, mainly, for the classical operations, namely, Create, Read, Update, Delete (CRUD) records. It is well-known that for each table, we create the corresponding DAO. Besides, from tables involved in the model of the enterprise, we can instantiate the module. However, MEMO I4.0 recommends to create a DAO per module, not per table to make the instantiation straightforward. In addition, in reality, the physical world may provide messages as a raw data format that needs a transformation before using it in the cyber world. So, again, we use the data access layer to integrate this raw data provided by smart resources from the physical to the cyber world.

In the case of E1, we need to create PlanDAO, ScheduleDAO, and AssemblyDAO objects, responsible for instantiating a Plan, Schedule, and Assembly modules, respectively.

4.4.3.4 Building the Business Model Layer

The business model layer is intrinsic to the system at study. For instance, we can instantiate the metamodels to obtain the model of the enterprise for a specific business logic. After that, we need to generate the business objects corresponding to the chosen business logic. To do so, we usually use the function data of the involved modules, i.e., those which contribute to this business logic. We resort to the data access layer to read these function data and build the business objects corresponding to the modules.

For example, Figure 4.15 illustrates two configurations of E1, one using lines and the other using cells. They are two business logics of the same enterprise. Furthermore, we can use the function data defined in Table 4.9 to build the business object of the Schedule module. Concretely, these function data will be considered as attributes of the *schedule* class.

4.4.3.5 Creation of the Mapping Layer

The main role of the mapping layer is to allow the instantiation of the model represented by the business layer to a runnable object in a specific plateform. Using such layer helps to respect the interoperability principle by ensuring the decoupling between the business object and the simulation object via the module object (see Figure 4.17). Thanks to the mapping layer, two main advantages can be highlighted. First, the flexibility of the development architecture, i.e., if the simulation platform must be changed, only the

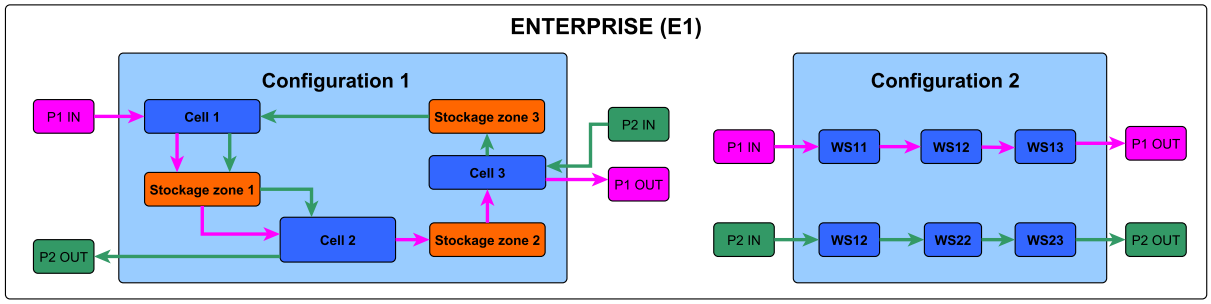

Figure 4.15: Illustration of agility principle using an example of plant configurations.

mapping layer must be adapted. Second, the time-saving, especially when the mapping layer is created by considering the architecture of the specific platform.

In the case of E1, to illustrate the business logic and generate a model instance, the business objects are represented by a Workstation, Cell, and Storage Zone, along with their attributes. However, the integration of this model with other systems requires also a mapping. For example, ERP or PLM systems are always external to the enterprise, and attributes defined in the E1 model are not necessarily defined in the same way in these provider systems. As explained in Subsection 4.4.3.3, we resort to the data access layer to consume information from these systems, then process them to map with the E1 model.

4.4.3.6 Writing of the Pseudo-Code of Modules

In addition to rules imposed by guidelines, we capture other rules derived from system reality using a Gemba walk and meetings with the team project. The pseudo-code format formalizes these rules and simplifies later coding steps to produce the simulation-optimization model.

In Appendix E.4, we provide the pseudo-codes corresponding to the functions of the process of E1.

4.4.3.7 Building an Optimization Model or a Decision Tree

Not necessarily all decisions related to the modules require an optimization model. It depends on the modeling objectives fixed with the managers and end-users. This step is conducted when a simulation-optimization-based or optimization-based DSS is expected. The goal is to formalize the decision(s) among those chosen by managers in Subsection 4.4.1.7, with the objective to optimize specific criteria.

In the case of E1, we decide to use a job shop optimization model based on the Equations (4.1)–(4.6) to improve the production schedule.

Let us consider a directed graph $G = (\mathcal{N}, \mathcal{A})$.

- $\mathcal{N}$: Nodes used to represent the set of operations performed on n jobs.

- $\mathcal{A}$: Conjunctive arcs represent the operating sequence of a single job, i.e. the precendence relationship to perform the operations.

Each node (i,j) denotes that job j is on machine i. P_{ij} is the processing time of the associated operation to node (i,j), and X_{ij} is the start time of this operation.

$$Minimize \;\; C_{max} \tag{4.1}$$

$$X_{kj} - X_{ij} \geq P_{ij} \;\; \forall (i,j) \rightarrow (k,j) \in \mathcal{A} \tag{4.2}$$

$$C_{max} \geq X_{ij} + P_{ij} \;\; \forall (i,j) \in \mathcal{N} \tag{4.3}$$

$$(M + P_{il})Y_{ijl} + (X_{ij} - X_{il}) \geq P_{il} \;\; \forall (i,l) \in \mathcal{N}, (i,j) \in \mathcal{N} \tag{4.4a}$$

$$Y_{ijl} + Y_{ilj} = 1 \tag{4.4b}$$

$$X_{ij} \geq 0 \;\; \forall (i,j) \in \mathcal{N} \tag{4.5}$$

$$Y_{ijl} \in \{0,1\} \tag{4.6}$$

The objective function of Equation (4.1) minimizes the makespan C_{max}. Equation (4.2) satisfies the machine i is prior to the machine k to perform the job j. The Equation (4.3) means that the completion time of each job j in each machine i does not exceed the maximum completion time. Equations (4.4a) and (4.4b) make sure that the job j and l could not be executed at the same time in machine i. Equation (4.5) indicates that the start time of each job j in all machines is positive. Finally, the Equation (4.6) defines a binary variable. M is a large number, used as an upper bound. Equation (4.5) indicates that the start time of each job j in all machines is positive. Finally, the Equation (4.6) defines a binary variable. M is a large number, used as an upper bound.

4.4.3.8 Choosing the Solving Approach

In this step, we choose how the optimization model will be solved. For instance, we could use an exact method and implement the optimization model in one of the existing solvers. Then, we inject the obtained solution into the simulation model in order to simulate it and test it under more realistic conditions, e.g., random MS behavior. Note that the choice of the solving approach depends also on the complexity of the problem instance of the company. In some cases, resorting to heuristic or metaheuristic approaches will be necessary. These latter can be integrated easily into the database system using stored procedures.

4.4.3.9 Conducting Numerical Experiments

In this step, we generate several instances to conduct numerical experiments on the model. Mainly, we use the historical data of the enterprise, because, the main goal is to check that the chosen solving approach is adapted to deal with the real instances of the enterprise. Furthermore, we document all tests and analysis results, and based on, the identified improvements of the solving approach could be implemented.

4.4.3.10 Building a Simulation Model of Modules

We use the Figure 4.16 to explain this step. Remark that it is important to perform the separation of the model of module into submodules to ensure the decoupling of the flows. Furthermore, the implementation of the business logic in the simulation software language is quasi-direct based on the pseudo-code of the module. Besides, the data available in the simulation model must allow to compute the KPIs defined based on their formula.

Note that the business objects corresponding to the module are generated according to the data model. The mapping layer ensure the transformation of the business objects to the simulation objects by using the module objects that are specific to the simulation software. Do not confuse the module objects that serve for the mapping with the module describing the function of the activity (e.g., Plan or Assembly).

When, we decide to change the simulation software, we might change the module objects dedicated for the mapping whether the language of the simulation platform change. However, the modeling choices remains the same.

Finally, the parallel, compatibility-driven development, and integration of modules are carried out, and the expected behavior of the integrated model of modules is verified.

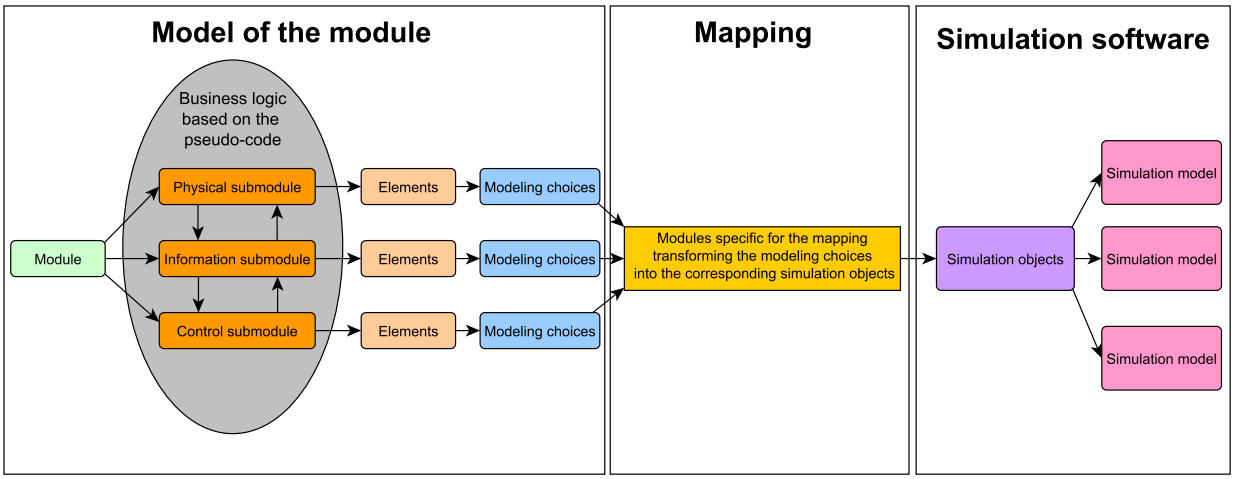

Figure 4.16: Approach to build the simulation model.

4.4.3.11 Mapping of the Decision Variables of the Optimization Model with the Simulation Model Attributes

This step is helpful when envisaging the possibility of integrating the simulation model with an optimization model. It allows to create a bridge between the decision model and information model. As shown in the partial integrated model of views (see Appendix C.1.1), all CIAOC properties of a decision are considered as an MS feature. Each element (see Subsection 4.4.1.13) has a set of attributes as we will see in Subsection 4.4.4.1. The attributes of all elements involved in the modules forming the model at study present the attributes of the simulation model. The goal is to map decision variables of an optimization model with the attributes of the simulation model builded by modules.

Table E.3 shows the mapping between the decision variable X_{ij} and the attributes (aRank, and aStartTime) of the product in each period that are used in the queue of the process element of the product entity.

4.4.3.12 Creation of Traceability Files

The use of traceability files allows easy debugging of the models of modules. The traceability file fills up at the same time as model execution. Therefore it can be used as a reference for comparison with the actual data generated when the process is executed. The attributes to track in the process must be included in this file.

For instance, in order to ease the model verification of E1, traceability files are created for each module that includes all its attributes and variables as illustrated in Table 4.15.

Table 4.15: Attributes of traceability file of each module of E1.

Plan	**Schedule**	**Assembly**
Replication	Replication	Replication
Period	Period	Period
Workcenter	Workcenter	Workcenter
Customer	Cell	Cell
Quantity estimated		Workstation
Product	Product	Product
Lead Time required	Shift	Resource
Planned Manufacturing order	Manufacturing order	Manufacturing order
Time of reception of demand	QuantityToProduce	Produced quantity
-	rank in the sequence	Operation
-	-	Start production time
-	-	End production time
-	-	Time of leaving the workstation
-	-	Time of arriving to the next workstation

4.4.3.13 Launching the Experimentations

The main goal of this step is to validate the simulation model behavior regarding the real-system behavior. We use the distributions of the input variables of the module, then, we run the simulation for a significant period w.r.t. the expected behavior. Finally, we perform the comparison of simulation results (output variables) of different replications with the historical data. Depending on the gap, corrections could be envisaged to fit better the real-system behavior.

4.4.3.14 Running the Corner Tests

In this step, we seek to ensure the stability of the model behavior. Therefore, we conduct the corner tests using the sensitivity analysis.

We launch several scenarios consisting to change the values of the input variables within defined boundaries and see how the values of the output variables are affected.

4.4.3.15 Displaying the KPIs

The DSS can communicate raw or aggregated information to compute the required KPIs of each module. Usually, we implement in the Performance Management System (PMS), the dashboards and models responsible for respectively displaying and processing these information (e.g., AI algorithms) to extract suitable KPIs and perform system analytics.

4.4.3.16 Checking the Module Agility and Flexibility

Experimentations are launched first to check whether the genericity degree of module fixed is respected, then to check if the module flexibility is achieved w.r.t. the chosen scale level. Often, we resort to various production scenarios to check the ability of model to handle diverse and dynamic MS environment.

For instance, below some scenarios that could be launched to test the genericity of the Assembly module of E1:

- Several cells in the workcenter
- Machine availabilities depending on the period
- Generation of many manufacturing orders in parallel

Likewise, we could test the flexibility of the Assembly module regarding the scale level fixed at the workstation level by launching the following test scenarios:

- Choose different production policies for workstations of the same cell
- Each workstation has its own production calendar
- Processing time of an operation depends on the workstation on which it will be performed.

4.4.4 Maintenance Phase: Versioning of the Model of Each Module and Generation of the Corresponding Documentation

The documentation related to each module is created to track modeling choices and manage module version by a versioning platform. All tests are run in the test branch; it is recommended to save all versions of the tests regardless of whether their exit status is positive or not. Then, based on the traceability files and KPIs, the results are discussed with the team in review meetings to validate or not the module version. In case of validation, and only after compatibility among views is checked, a versioning commit is performed. Otherwise, an analysis of how the model can be adjusted must be conducted in the design phase, e.g., by changing the control logic or the scale level. For example, suppose the degree of genericity decided at the definition phase is no longer correct to model the process, therefore, we update the module or create a new one from scratch.

Note that we follow a continuous improvement approach to develop all functionalities of the module, i.e., we perform the four phases of the strategical/tactical stage (definition, design, implementation, and maintenance) each time a new module version including new functionalities is required and asked from the director/managers of the enterprise.

4.4.4.1 Documentation and Versioning of Module Models

In this step, according to the modeling choices made at the design phase, we generate the detail of each element involved in the module, i.e., its attributes and their descirption. Besides, the versioning of the SADT diagram is required. In addition, documentation of each module version must be kept up to date, i.e., modeling choices, models of each view, simulation models, optimization models, traceability files and experimentations for validation of models.

Note that we use a versioning plateform (e.g., git) to keep the module versions and their associated documentation updated.

4.4.4.2 Checking the Compatibility Among Views Models of Modules

In this step, since the models of views of the module is derived from the metamodels, and we already demonstrate the integration of these metamodels, therefore, the model of the module obtained is compatible, if OCL constraints are consistent. However, the integrated model of modules must be verified. The incompatibility between one module requirements and another one is checked based on the data model and the OCL constraints.

4.4.4.3 Organization of Review Meetings with the Project Team

The goal of the review meetings with the team project is to validate or not the module version according to the requirements defined at the definition phase. It is also, the time to

correct the modeling choices and adapt the modeling approach if needed. We recommend using the animation for the verification of the simulation model behavior regarding the real system. Based on KPIs, we decide to validate or not the module version. It is also the occasion to plan the development of new functionalities of the module version.

4.4.4.4 Updating Documentation and Versioning of Models of Modules

MEMO I4.0 is based on a continuous improvement approach to improve the functionalities of modules and validate them after each improvement. When a new module version is released, we carry out an update of module documentation, and we perform a new commit in the main branch of development to ensure the availability of the last version of module in the library modules.

In the case of E1, according to the evolution of the production activity, either new functionalities must be added to the existing modules or new modules (functions) are developed based on the same principle.

4.5 Application of MEMO I4.0: Operational Stage

The operational stage of MEMO I4.0 consists of using the developed modules in the strategical/tactical stage to meet the needs of the specific use case. First, we specify the required modules, then we build the integrated model using the models of modules, after that, we instantiate it and validate the obtained results, and finally, we provide the insights to the directors/managers to help them in their decisions.

4.5.1 Definition Phase

Starting from the needs expressed by the management, the modeler decides whether they can be covered with modules that have been already derived by the DNA/guidelines of the group; if not, modules are designed from scratch in a classic scheme. In the latter case, after choosing the suitable preexisting modules, the modeler decides on the new ones required to generate the integrated model.

The creation of new modules must follow the same development described in Section 4.4 to put it in a strategic/tactical perspective. However, to avoid the creation of new modules as much as possible, it is suggested to decide the genericity degree and the scale level to consider expected evolutions.

4.5.2 Design phase

Once all the required modules are available, the modeler checks whether they need to be updated. If so, this change is made in the maintenance phase of the strategical/tactical stage, along with the related documentation. Then, the modeler assembles these modules

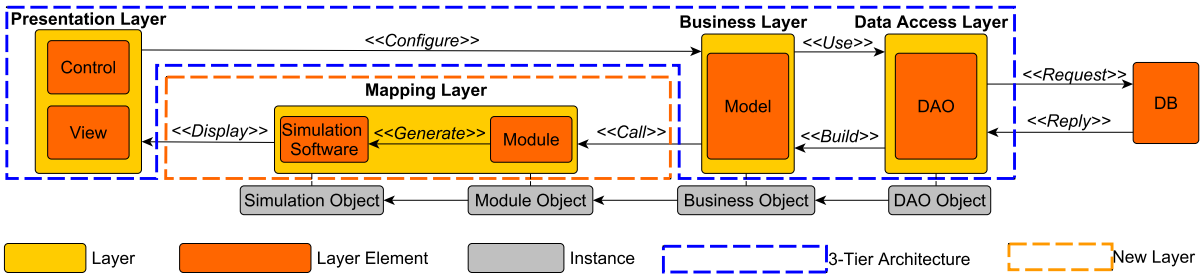

Figure 4.17: The Development architecture for the integrated simulation model.

to obtain an integrated simulation/optimization model, whose behavior is already checked at the implementation phase of the strategical/tactical stage.

A version of the integrated model can then be instantiated to simulate a specific configuration of the MS and perform a what-if analysis. Therefore, we instantiate the models defined in the design phase of the strategical/tactical stage, to be precise in the step instantiation of the meta-models of views.

Finally, to comply with the Interoperability principle of MEMO I4.0 (see Subsection 4.3.2.3), a mapping layer of the architecture of the integrated model Figure 4.17 is defined that depends on the specific simulation platform and transforms the business model into a runnable simulation object.

4.5.3 Implementation phase

We use a flexible development architecture inspired by classical 3-tier software architecture as illustrated by Figure 4.17. It consists of a presentation layer (user interface), a business layer (logic), and a data access layer. We add a mapping layer to be able to instantiate the integrated model of modules, represented by the business layer, to a runnable object. The tuning of the integrated model settings, which is needed for the mapping layer to derive the simulation object, can be partially automated and fed by the MES in some cases (e.g., with digital twins).

In Figure 4.17, we present the mapping layer only for the case of a simulation model. Still, the same goes for the optimization model, we only need to adapt the mapping layer for the optimization software. It means the optimization model will use the same business layer as the simulation model. Optimization and simulation can be combined according to the approaches described in Subsection 1.4.4.

The validation step is based on historical data, Gemba walk, and experts' advice. Two situations (As-Is or To-Be) are possible in the validation process. In the former, the simulation/optimization model represents the existing system: the validation can then be based on the Gemba walk and meeting review to discuss the KPI values with experts. Historical data and statistical tests (e.g., chi-square goodness-of-fit) can be used to study

the difference between the observed and the expected values. As for the To-Be situation, the simulation/optimization model represents a future system, and the validation cannot be based on historical data: the correctness of model behavior is then ensured based not only on expertise but also on sensitivity tests and other statistical tests (variance analysis, regression analysis).

4.5.4 Maintenance Phase

The same actions of the development of modules are performed, except that versioning is made for the integrated model. The documentation related to the process and simulation/optimization model, including solutions and recommendations, is created or updated. This task has the highest priority in allowing knowledge capitalization and version improvement.

4.6 Conclusion

In this chapter, first, we investigate the evolution of EM modeling, particularly the EM principles and dimensions within Industry 4.0. The analysis conducted shows that a strong consistency exists between them. The ultimate goal of using the EM dimensions and respecting the key principles is to achieve the modeling objectives imposed by Industry 4.0 scenario (see Table 4.1).

Afterward, we propose to define useful concepts for conducting the modeling activity, especially, to explain how the genericity degree must be chosen and suggest three possible approaches depending on the case to model and the expected genericity degree. Then, we found that additional views than those already defined are required to avoid interoperability issues. We detail each view to demonstrate how it contributes during the life cycle of model building. Next, we succeed in integrating all conceptual models of each view.

We propose a new methodology of enterprise modeling oriented Industry 4.0 (MEMO I4.0) that respects the key principles and dimensions already identified. We noted that using only the life cycle and view dimensions allows for obtaining a consistent and modular model. However, it would be struggling to evolve the developed model since modeling choices about the granularity and the genericity have not been considered at the early stage. In other words, if we have a new requirement, we need to start from scratch and spend a lot of time on the model development. Therefore, we decided to add the genericity and scale dimensions to achieve agility and flexibility during the modeling process. Furthermore, we detail the structured approach of MEMO I4.0, which describes the modeling process from the definition of requirements until the implementation and maintenance of models.

Now, as we understand the MEMO I4.0 framework and its principles, and we provide the structured approach accompanied with the roadmap to apply MEMO I4.0, we propose using MEMO I4.0 for a real-case in the next chapter.

Chapter 5

Application of MEMO I4.0 to Real Industrial Environment: a Case Study

Résumé

Ce chapitre applique la méthodologie MEMO I4.0 dans un environnement industriel réel de l'entreprise elm.leblanc, spécialisée dans la production de systèmes d'eau chaude. Un cas d'étude parmi d'autres est présenté dans ce chapitre pour soutenir l'entreprise dans sa transition numérique. MEMO I4.0 est appliqué dans un scénario brownfield puisque l'entreprise dispose déjà d'un système de production. Trois nouveaux modules sont développés (Préparer, Charger et Transporter) pour répondre aux besoins des managers concernant l'impact de la politique d'alimentation des lignes sur la performance de la logistique interne. Dans ce premier cas d'étude, une approche de simulation est utilisée pour évaluer les performances logistiques internes. La méthodologie MEMO I4.0 a permis de fournir des informations pertinentes aux managers en offrant des réponses rapides et efficaces. Les résultats ont montré l'efficacité de cette approche dans un environnement industriel incertain où l'agilité est essentielle. Deux autres cas d'études sont présentés en Annexe. Un deuxième cas d'étude, où un module supplémentaire (Produire) est développé et intégré avec les modules existants pour trouver la configuration appropriée de l'usine face à une demande croissante. Enfin, dans le troisième cas d'étude, le module Produire est mis à jour pour inclure un nouvel indicateur clé de performance lié à l'ergonomie, et un système d'aide à la décision basé sur un couplage de la simulation et l'optimisation est développé pour résoudre le problème de conception de ligne de production. L'utilisation de la capitalisation des connaissances dans MEMO I4.0 a considérablement réduit le temps de développement des systèmes d'aide à la décision adaptés à l'environnement de l'Industrie 4.0.

Contents

5.1 Introduction 184
5.2 Strategical/Tactical Stage: Development of Modules 185
5.2.1 Definition Phase: Requirements and Specifications of Modeling 187
5.2.2 Design Phase: Instantiating Metamodels of Views . . . 197
5.2.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module 205
5.2.4 Maintenance Phase: Versioning of the Model of Each Module and Generating the Corresponding Documentation 215
5.3 Operational Stage: Investigating of the Internal Logistic by Simulation 216
5.3.1 Definition Phase 217
5.3.2 Design Phase 217
5.3.3 Implementation Phase 218
5.3.4 Maintenance Phase 219
5.4 Conclusion 220

5.1 Introduction

In the previous chapters, we tried to provide answer to the research questions defined in Chapter 1, ultimately concerning the features that an Enterprise Modeling Framework should have in order to be capable, based on an holistic view of Manufacturing Systems (MSs) to derive Industry 4.0-oriented Decision Support Systems. The result was the definition of a new methodology, MEMO I4.0, that allows modeling an MS in the Industry 4.0 context.

In this chapter, we will apply MEMO I4.0 in a real industrial environment, represented by elm.leblanc, a French company that belongs to the thermo-technology division of the Bosch Group, specializing in the manufacturing of hot water systems, most notably boilers. The application of MEMO I4.0 is meant to support elm.leblanc in its numerical transition. In doing so, and since an MS already exists, a brownfield scenario has been dealt with. This gave rise to a complex case study that we report here in detail.

As we explained in Chapter 4, in the structured approach of MEMO I4.0 the trigger of the module development is the use case expressed by the managers. In the case of elm.leblanc, the use case consisted of investigating the internal logistics and understanding the parameters impacting the logistic efficiency of one of the plants of elm.leblanc. This use case led to the development of three new modules (prepare, load, and transport), according to the roadmap described in the strategical/tactical stage (see Section 4.4), which were then added to the library of modules to allow reuse.

Two more case studies have indeed been dealt with, which for the sake of brevity we

do not repore here in detail, but in Appendices F.5 and F.6 instead. The second use case tries to handle an increasing customer demand and seeks to find the appropriate plant configuration to face such a situation. In this use case, we only needed to develop one new module (Produce) and integrate it with the three modules previously inserted in the library of modules. Finally, a third use case has been dealt with where a line design problem has to be solved which considers a KPI, ergonomics, which is emerging with the advent of Industry 4.0. After analyzing the manager needs, we found that the "Produce" module must be updated to include the new KPI, before being coupled with an ALBP optimization model (see Appendix F.6) to obtain a solution that can be simulated under real conditions. After each use case, we enrich and update the library of modules as shown in the Figure 5.1. By doing so, we reduce the development time considerably.

Figure 5.1 also shows that for each case study, we started at the operational stage that was described in Section 4.5. First, we analyzed the manager needs, then decided which new modules were required, and we developed them in the strategical/tactical stage. After that, we went back to the operational stage for the instantiation of modules and the performance assessment (see Section 5.3 and appendices F.5 and F.6). For the sake of simplicity we omitted the description of the strategical/tactical stage three times and presented the development of all modules in just one time, as if we knew in advance what were the modules required for the three use cases. Based on this, Section 5.2 is interested in the strategical/tactical stage of developing all modules required for the MS modeling in the three use cases respecting the life cycle of MEMO I4.0. Next, Section 5.3 deals with the operational stage, where the first use case asked by managers will be detailed. Finally, Section 5.4 concludes this chapter.

5.2 Strategical/Tactical Stage: Development of Modules

Before implementing a DSS, we need to determine which DSS type is the most fitting. In Subsection 1.4.2, we identified different types of DSS in terms of applications, with model-driven and data-driven DSSs being the most commonly adopted in smart factories. Data-driven DSSs are often resorted to since data integration and standardization of data exchange allow to develop them quickly (see Subsection 2.3.1.1), while model-driven DSSs allow to deal with the complexity of a smart factory induced by the vertical, horizontal, and end-to-end integrations of it (see Subsection 1.4.4). However, our analysis reveals that to exploit the potential offered by a smart factory, a combination of the two types of DSS is to be preferred. Hence, we propose developing a model-driven DSS while using a fully data-driven model. By doing so, we also guarantee a reduction in development time.

DSS must result from a new EM framework, such as MEMO I4.0, to advance data integration and processing possibilities. Therefore, in the context of the case study

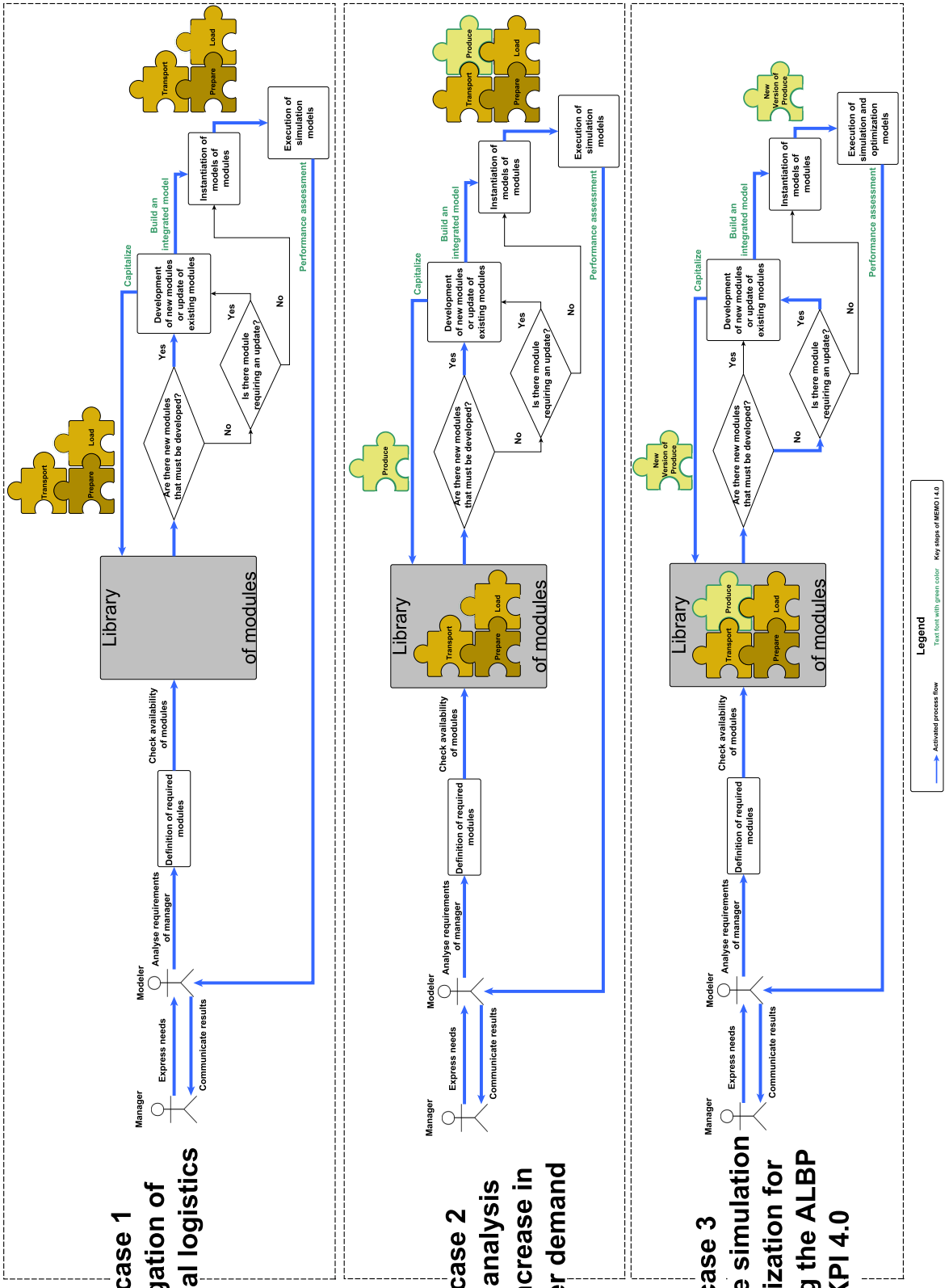

Figure 5.1: The use cases within elm.leblanc handled by MEMO I4.0, and the chronology of modules development and capitalization.

described in this Chapter, we choose to develop a model-driven DSS based either on simulation or simulation-optimization approaches.

5.2.1 Definition Phase: Requirements and Specifications of Modeling

In this phase, we will follow the steps defined in Chapter 4. The main goal is to delimit the scope of the study, identify the modules required, and specify the requirements w.r.t. the vision, DNA, and guidelines of elm.leblanc.

5.2.1.1 Definition of Modeling Objectives and Scope

Several brainstorming sessions with the team project were required to specify the intended objectives of the modeling work. This modeling step helps the enterprise understanding its system behavior first, then answering questions mainly linked to future transformations. The main objectives expressed by managers and directors were:

1. Testing different layouts to improve the internal logistics of the plant.
2. Limiting stock between a lower and an upper bound.
3. Defining inventory policies for holding a necessary stock to meet customer demand requirements.
4. Identifying the impacts of external elements on the performance of the plant.
5. Taking into account the variety of finished products.
6. Assessing the impact of transforming current equipment into agile equipment.

Based on the modeling objectives, we decided to delimit the scope of modeling to one specific elm.leblanc plant.

5.2.1.2 Description of the Enterprise Vision, Guidelines & DNA

Before providing the enterprise vision, we made a SWOT analysis with the direction department and validate it with them, which we report in Figure 5.2.

From SWOT, we could deduce that introducing new products in new or existing lines is frequent. Thus, the number of lines, their workstations, and the number of products assigned to each line must be an input parameter. Our interpretation of the SWOT analysis regarding modeling requirements is as follows:

1. Reconfiguration times must be reduced, so we must develop an MS to be an RMS.

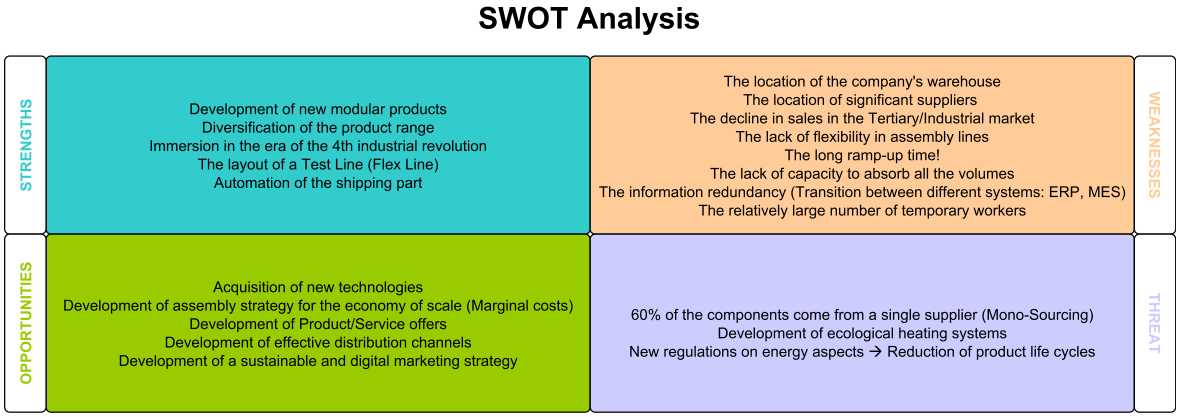

Figure 5.2: The SWOT analysis to guide the elm.leblanc vision.

2. A data integration strategy is necessary to avoid redundancy and achieve vertical integration.
3. Multiple-suppliers are considered to provide the components and avoid shortages, which allows to assume no inventory shortage in the warehouse.
4. The flexibility of assembly lines should be able to switch from a push strategy to hybrid strategies so as to achieve product customization and time to market reduction.
5. The number of temporary workers is increasing, impacting both the production cost and the rework time.
6. The acquisition of new technologies requires to take into consideration different types of workstations: manual, semi-automatic, and automatic.
7. The process of an operation can be done with one or several resources.

As regards the enterprise guidelines, which are based on the Bosch Production System (BPS), we have agreed not to share them for confidentiality reasons. However, based on interviews with the experts and several Gemba walks, a list of best practices that can impact modeling choices in the design phase, although non-exhaustive, can be given as follows:

- Once an order is placed, the whoesalers is guaranteed to be delivered the following day.
- The same happens for the delivery of spare parts to convenience stores.
- All unit measures of the orders are fixed at "*ea*" since we are in the case of discrete manufacturing.

- The production leveling is done weekly, taking into account available staff, available stock, and production forecasts. The leveling is a methodology used to achieve a balanced utilization of resources as well as a constant flow and rhythm along the entire value stream, even in the event of fluctuations in customer demand.

- The lines are U-shaped and are dedicated to a certain product family.

- The supermarkets are recommended for line feeding.

- The appropriate feeding policy depends on the class of products: A (runner), B (Standard), and C(Exotic). The terms runner, standard, and exotic refers respectively to the high, medium, and low quantity of units within a specified period.

- Product families can have different assembly sequences.

- The workstations have a common base but differ in types of equipment.

- For each product family, a diagram is used to illustrate the tasks processing time, the precendence constraints among tasks, the quantity of components to consume in each task, and the task assignment restrictions to the workstations.

- The MTM2 standard is used, which is a method to analyze work processes and determine planned and standard work times in order to describe, structure, design and plan work systems.

5.2.1.3 Decomposition into Modules and Definition of their Functional Description and Assumptions

An SADT diagram given in (Figure 5.3) is used to describe the functions (modules) included in MS activities, namely, Prepare, Load, Transport, and Produce. The decomposition of elm.leblanc plant into three macro-activities (production, feeding, and storage) presented in Chapter 2 matches the manager vision and department organization. Note that the finished goods are transferred to the expedition area using an AGV solution. Our study intentionally omitted this transfer function as it fell outside the scope of the elm.leblanc priorities.

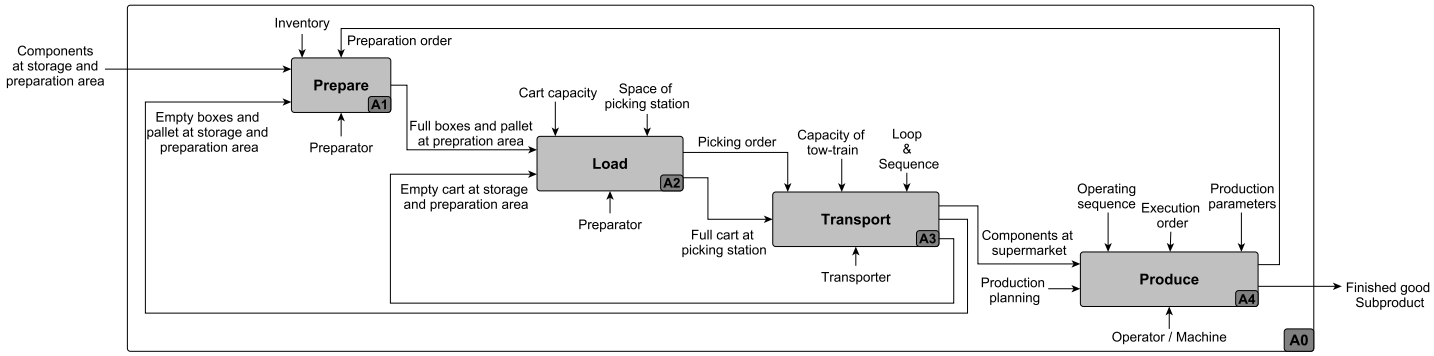

Figure 5.3: SADT of MS process.

Once functions involved in the MS are identified, the business logic corresponding to their processes must be captured. Several tools, such as textual description, flow chart, process mapping, and BPMN, can be used for this purpose. In our case study, a textual description was preferred since it is simple to use, i.e., no specific technical skills are necessary to have a common understanding of the business logic among project team members. When it was required, flow charts were used to detail some processes with the technical team. MEMO I4.0 does not recommend using a specific tool for studying business logic but states that the tool choice is project-specific.

Before introducing the assumptions made for each module, a short description of the plant activities can be useful. The plant has two storage areas and several assembly lines. Each line can produce one or more products according to a push policy between workstations. Each storage area has its own picking station to store full carts when prepared. Milkruns load full carts and follow specific routes to feed the assembly line with components. They stop at each supermarket, unload the corresponding boxes and carts, retrieve empty ones and go back to the picking stations to drop them.

The consumption of components in supermarkets by workstations depends on the tasks and the product bill of materials. A preparation order is sent to the associated storage area for replenishment either when the trigger quantity is reached or when the product lot size is assembled. Preparation orders are fulfilled according to some priority rules.

The location of storage areas, picking stations, supermarkets, and workstations are given and must be respected. The picking station space, the picking capacity of carts, and the storage capacity of supermarkets are given as well.

In each shift, a break time should be allocated. Each day, the logistic manager decides the resources (preparators and milkruns) to allocate to each line, while the production manager defines production planning and decides the resources (operators) assigned to each workstation. In addition, based on technological constraints and for optimization reasons, the production sequence of the products is fixed.

Table 5.1 provides the assumptions identified for each module; some of them are derived from the enterprise vision (e.g., no inventory shortage), others from the real-life of the studied plant (e.g., only one central warehouse), and others are assumed for simplifying the development (e.g., constant speed of the vehicle). Finally, based on the notation already defined of MS features (see Subsection 3.4.4, and Appendix B), an ID of each assumption is given.

Table 5.1: Assumptions related to each module.

Module	Designation	Code
Prepare	Bins are available at the beginning of Planning Horizon	R.a.34
	Only one central Warehouse	R.a.30
	Warehouse has enough parts to supply demand	R.a.31
	Similar pickers skills	R.a.49
	Finite number of pickers	R.a.52
	Standard pick time for all zones	T.a.26
Load	Number and destination of bins are known	R.a.36
	Stochastic loading & unloading times	T.a.15
Transport	Every location has the same size	R.a.50
	Tow train capacity limited only by number of bins	R.a.57
	Each route is assigned to one vehicle	P.a.14
	Constant speed of vehicle	R.a.23
Produce	Task times are known	T.a.9
	A task can be assigned to only one Workstation	P.a.7
	Production sequence is known	P.a.11
	Task assignment restriction to machine	P.a.8
	Several products can be manufactured in period t	P.a.16

5.2.1.4 Separation of Flows and Definition of Submodules

For each function, the flows (physical, information, control) are separated to comply with the modularity key principle of MEMO I4.0. Table 5.2 shows this. Then, we define submodules of each module using the direct mapping with the flows identified in Table 5.2: this is shown in Table 5.3. As we said in Subsection 4.4.1.4, each module corresponds to one function and is subdivided into submodules. Each submodule corresponds to only one type of flow.

5.2.1.5 Building Modeling Scenarios

In this section, we introduce the reader to the aspects that need to be integrated in each submodule.

The preparation process uses available preparators in the storage area to retrieve components requested by the preparation orders. Moreover, the preparation time

Table 5.2: Flows of each function of MS activity.

Flow type	Prepare	Load	Transport	Produce
Physical	Component, Box, Pallet	Box, Pallet, Cart	Milkrun driver, Cart	WIP
Information	Preparation order	-	Picking order	Execution/ Fabrication/Launch/ Transfer orders
Control	Preparation order control, Inventory, Box check	-	Picking order Control , Check empty boxes/carts	Manufacturing order control, demand control
	Shift/resources control			

Table 5.3: Submodules corresponding to modules.

Submodule/Module	Prepare	Load	Transport	Produce
Physical submodule	Preparation process	Loading Process	Transport process	Production process
Information submodule	Preparation orders	-	Picking order	Manufacturing order
Control submodule	Prepartion control	-	Transport control	Production control
	Shifts/resources control			

depends on the size of the picking list and the component storage support, while The preparation order depends on the component inventory in each supermarket. The submodules devoted to the preparation process and preparation orders must then take these elements into account. As for the control preparation submodule, it is responsible for updating the component inventory, which triggers the generation of the preparation orders according to the line feeding policy. In addition, it controls the availability of storage support in each storage area, and it assigns the preparator to the picking list.

In a similar way we build the loading process submodule. Indeed, the loading process uses available preparators to load the full storage support in available carts at the picking station, while the loading time depends on the storage support, and each storage area has its own cart type. If carts contain several stages, the preparator loads carts according to the preferred order of stages.

Concerning the transport process, the following aspects must be integrated in the corresponding submodule. Depending on the picking orders, the transporters organize their routes to retrieve different carts from the picking stations and feed the lines via the supermarkets. Generally, the routing decision defined in the supply activity is solved when dealing with this process. Meanwhile, the picking order submodule takes care of the information flow and generates orders depending on the number of full carts at the picking stations, while the control transport submodule manages the control flow and assigns transporters to the picking order according to the picking strategy (single order picking, batch picking, zone picking, wave picking, and bucket brigade picking as recalled in Subsection 2.2.3.4). The latter submodule also manages the empty boxes and carts after the line feeding.

Concerning the produce function, the production process submodule is responsible of deploying the available resources (operators, operators and machines, or only machines)

depending on the production needs to perform tasks, taking into account that the production can be organized either in lines or in cells. The production time then depends on the product, the task, and the resources used. The manufacturing order submodule generates the orders according to the production planning (i.e., the product quantity to produce each day) and the production sequence. The production control submodule manages the control flow according to the production strategy (push, pull, and hybrid, see Subsection 2.2.1.1) and controls the assignment of production resources to workstations.

Finally, the shift/resource control submodule manages the daily shift organization according to the schedule; it also allocates the capacity of preparators for the Prepare and Load modules, the capacity of transporter for the Transport module, and the production resource capacity for the Produce module.

After giving specifications for each submodule, scenarios can be built to study the current (As-Is) and future (To-Be) situation. We refer the reader to Appendix F.1.1 for a detailed presentation of the scenarios related to all the mentioned submodules. These scenarios are later used in the design phase to specify the most-fitting approach for the modeling of the submodules according to the expected genericity level.

5.2.1.6 Specification of the Modeling Context and the Genericity Degree

Based on the previously built scenarios, we decide to focus on the As-Is situation because managers aim to understand which factors have an impact on the performance of internal logistics performance and what are the effects of an increasing demand on line efficiency. This would allow, based on the analysis of the results, to determine viable improvement paths. For the Prepare and Load modules, we fix the genericity degree to *specific*, and for the Transport and Produce modules to *partial*. This choice is motivated by two main reasons. First, the Bosch Production System guidelines provide few details on the preparation and loading functions compared to the other functions, therefore only the current preparation and loading processes of elm.leblanc can be considered as a reference for the development. Second, experts confirm that future decisions concerning the MS evolutions will most probably focus on the last two functions. For these reasons, we decided to put greater efforts into increasing the genericity degree of the Transport and Produce modules.

5.2.1.7 Identification of Decisions related to Modules

Based on the objectives defined in Subsection 5.2.1.1, a list of decisions corresponding to each module can be derived, which is given in Table 5.4.

Table 5.4: Decisions linked to each module.

Prepare	Load	Transport	Produce
Line feeding policy	Inventory Planning & Control	Routing (Supply)	Capacity Investment Line Design

5.2.1.8 Definition of Module KPIs

KPIs for each module (see Table 5.5) were defined with the team project and used as assessment tools for our models.

Table 5.5: KPI of each function.

Prepare	Load	Transport	Produce
• Mean time to prepare one preparation order per storage area.	• Mean number of carts loaded per loop per milkrun driver.	• Mean number of loop per shift.	• Productivity.
• Mean time to prepare a list of preparation orders per storage area.	• % of carts filling.	• Mean number of empty boxes and carts (un)loaded per supermarket.	• Mean number of breakdown due to lack of components.
• per-shift usage % of preparators.	• per-shift usage % picking stations.	• per-shift usage % of transporters.	• per-shift usage % of operators/machines.

Some chosen KPIs are process-related, e.g., productivity; others are product-related, e.g., mean time to prepare one preparation order per storage area; and others rather resource-related, e.g., per-shift usage % of preparators.

5.2.1.9 Specification of the Scale Level

In Subsection 5.2.1.3, we chose to decompose the activities into main functions in accord with the roles of different actors. Each main function corresponds to one module, which is subdivided into submodules to have maximum behavior flexibility.

As we have seen in Subsection 4.4.1.9, it is recommended to use the information part for specifying the scale level of the module because it is the interface between the physical part and the control part of the module. The Preparation order, Picking order, and Manufacturing order submodule specify the scale level for the Prepare, Transport, and Produce modules, respectively. Besides, the Loading Process submodule is responsible for specifying the scale level of the Load module.

Furthermore, regarding the modeling objectives (e.g., considering the variety of finished products) and decisions to make (e.g., Inventory planning & control), it becomes clear that we need to fix the scale of all the modules at the *product* level.

5.2.1.10 Definition of Departments linked to Modules

Eight departments are concerned with modules and will contribute during the development life cycle. Some departments are involved in several modules, while others are involved in only one:

- the direction department participates in the definition of requirements and assessment of modules.
- the IS (Information System) department is critical in ensuring data integration among EIS for different modules.
- the HSE department contributes to the definition of the prepare module by specifying the storage support dimensions and weight. For the load module, the HSE (Hygiene Security Environment) specifies the loading preference order stage since, depending on the stage, the preparator posture and the injury risk will differ, while for the transport module, it indicates the maximum number of carts that could be towed and the maximum velocity of the milkrun driver. Finally, for the produce module, as can be seen in Appendix F.6, HSE makes the ergonomic study of operation execution at each workstation in order to specify the workload of tasks.
- the logistics department manages data related to the prepare, load, and transport modules, and the remaining departments focus on the produce module. Besides, the engineering department and process planning & maintenance department step also in the prepare and transport modules, respectively.

Table 5.6: Departments involved in each module in terms of data management.

<table>
<tr><th>Activity</th><th colspan="2">Storage</th><th colspan="2">Line feeding</th><th>Production</th></tr>
<tr><td>Function</td><td colspan="2">Prepare</td><td>Load</td><td>Transport</td><td>Produce</td></tr>
<tr><td rowspan="6">Department</td><td colspan="3" rowspan="2">Logistics</td><td colspan="2">Manufacturing</td></tr>
<tr><td colspan="2" rowspan="2">process planning & Maintenance</td></tr>
<tr><td colspan="2">Engineering</td><td colspan="2">Engineering</td></tr>
<tr><td colspan="3"></td><td colspan="2">Quality</td></tr>
<tr><td colspan="5">HSE</td></tr>
<tr><td colspan="5">IS</td></tr>
<tr><td colspan="5">Direction</td></tr>
</table>

5.2.1.11 Definition and Association of Information of Modules with MS feature

For each module, we specify all information required for modeling the plant of elm.leblanc. For the sake of brevity, we do not report the information of all modules

here but in Appendix F.1.2. For instance, Table F.12 provides the information of the Prepare module, which is decomposed into four function data (PreparationParameters, PreparatorsData, InventoryData, ComponentsData). The function data serves to group information corresponding to a function so that they fit better the business logic, while the data set collection emphasizes a grouping depending on the data source. For example, the function data "PreparationParameters" includes information that allow to modify the process parameters of the Prepare module. In most cases, for each information of the function data, an MS feature (as defined in Appendix B) can be associated, which means that these information is standardized and expressed in generic way. However, in specific cases, an information cannot be associated to an MS feature because it is specific to the business process of the studied plant, or it is only used for simulation model tuning. In such case, an information will be considered as a "*setting*". Blue color background is used in the function data tables of Appendix F.1.2 to distinguish the settings rapidly as plant-specific features.

5.2.1.12 Identification of the Raw Data and per-Department Organization

The data set collection is obtained from enterprise information systems or messages. These data set collections are considered raw data because, most of the time, they need to be transformed to become information. As discussed in Subsection 4.4.1.12, four main categories of raw data are considered: transactional data, master data, reference data, and metadata. In our case study, the main transactional data are related to the release of new preparation orders, new picking orders, or new manufacturing orders. The master data involves, among others, customers, products, resources, and operating sequences. Each master data must be managed, i.e., created, updated, and stored by one of the eight departments already defined. On the other hand, the reference data, such as the status of released orders, serve for real-time monitoring. Finally, the metadata provides contextualization and makes easy traceability, e.g., by providing the data source.

Appendix F.1.3 provides a list of data set collection of the elm.leblanc modules, their sources, and departments responsible for their management.

5.2.1.13 Identification of Elements Involved in Each Module

After determining the information and data required by a module, it is straightforward to identify the plant elements involved in it. Table 5.7 summarizes the plant elements involved in each module. For instance, the information *Preparator list in the storage area / Line / Date / Shift* indicates that a *Preparator* (first element) depends on the *Storage area* (second element) where he is assigned. Furthermore, the preparation process in each storage area is organized w.r.t. the Line (Third element) that requests the components. Two other elements impact the assignment of the preparator to a specific storage area for a specific line, namely, the *Date* and *Shift* elements. In a later stage, starting from the

elements associated with modules, we will determine those associated with submodules.

Table 5.7: Elements of plant included into each module.

Elements	Prepare	Load	Transport	Produce
Workcenter/Line	x	x	x	x
Workstation	x		x	x
Product	x			x
BOM	x			x
Component	x			x
Operation				x
Process	x	x	x	x
Storage Area	x	x		
Resource	x	x		x
Shift	x	x	x	x
Date	x	x	x	x
Schedule	x	x	x	x
Transporter			x	
Picking station		x	x	
Other Stations		x	x	
Supermarket	x		x	x
Storage support	x	x	x	
Cart		x	x	
Stage		x		

5.2.2 Design Phase: Instantiating Metamodels of Views

In this phase, we instantiate a metamodel for each view based on what we already decided in the definition phase. We also make the modeling choices related to the elements involved in each submodule, as well as to the modeling approaches that will be used, i.e., independent, intermediate, or integrated. At the end of this phase, MEMO I4.0 ensures the model development to be platform-independent. In other words, all models that will be specified will be fully reusable, as well as the modeling choices and approaches.

5.2.2.1 Decision of the Modeling Choices of Submodules

The next step concerns the decision over the modeling choice of elements of each submodule. As an example, we provide details for the Prepare module, whose

submodules are the Preparation Process, the Preparation Order, and the Preparation Control submodules. We also provide the Shift/resources Control submodule, which is shared among all modules.

We consider two things to determine the elements involved in each submodule of a module. First, we look at the elements involved in the module, which are given by columns containing "X" in Table 5.7. Second, we consider the modeling granularity we want for the module, i.e., the scale level fixed. The scale level is set to the *product level* for the Prepare module. It may be helpful to also resort to the description of different flows (physical, information, and control) that match with the submodules of the module. These flows help to select the elements involved and check if they are compatible with the elements involved in the module. Once we have determined these elements for each submodule, we decide how to model them.

Table 5.8 shows that the preparation order that triggers the Prepare module is generated for the specific component belonging to the set elements " *Workcenter/Line, Workstation, Product*". Note that we consider an attribute called "aCommonIndex", which represents an aggregation ID of "Workcenter/Line ID, Workstation ID, and Product ID". Since we fixed the modeling scale at the *product level*, using the "aCommonIndex" allows maximum flexibility to adapt the modeling granularity quickly. The preparation order can be customized by product, workstation, or workcenter/line. This allows for a unique preparation order for each product or a shared order for all products passing through the same workstation or workcenter/line. We manage inventory at the workstation level (element set "Workcenter/Line, Workstation)" since in one workstation, the same component can be used for multiple products. As a result, we cannot model the inventory as belonging to a specific product. Since we share the storage supports among all lines during the preparation process, we decide to manage them at the *workcenter* level. The preparators are dedicated to lines, but in some cases, they can be assigned to two specific lines. Thus, we manage them at the *line* level.

Depending on the line and the storage area specified in the preparation order, the preparation process will pick one of the preparators dedicated to the line and assign it to this storage area. Then, the preparation time depends on the storage support of the component and the quantity to be prepared. Finally, we handle shifts at the *workcenter* level and their schedule at the *line* level because depending on the line, we could have at one date one or more shifts.

5.2.2.2 Choice of the Modeling Approach of Each Submodule

Once modeling choices are made, now, a decision must be taken as regards the modeling approaches of each submodule. As already explained in Section 4.2, the approach depends on the specificity of the case at hand. The genericity degree for the Prepare module set to *specific*, in order to specifically focus on the As-I situation.

Table 5.8: Modeling choices of the elements involved in the submodules of the prepare module.

Modeling choice	Module	Prepare			All modules
	Element / Submodule	*Preparation process*	*Preparation order*	*Control preparation*	*Control shifts & resources*
Entity	Component	x			
Entity	Storage support	x			
Set	BOM	x			
Resource	Preparator	x			
Delay (Preparation time / Storage support)	Process	x			
Attributes of the entity preparation order	Workcenter/Line		x		
	Workstation		x		
	Product		x		
	Supermarket		x		
	Component		x		
	Storage Area		x		
Attributes of the entity Inventory to control the preparation order	Workcenter/Line			x	
	Supermarket			x	
	Component			x	
Attributes of the entity box check (storage support)	Workcenter			x	
	Storage support			x	
Attributes of the entity control capacity warehouse resources	Workcenter/Line				x
	Resource (Preparator)				x
	Date				x
	Shift				x
	Schedule				x
Entity	Date				x
Entity	Shift				x
Set	Schedule				x

As to the management of the Preparation Process submodule, current and future situations, discussed with the managers, indicate that the number of storage areas will not change. Furthermore, we use a dedicated strategy for the advanced area, while a shared strategy is adopted for the main warehouse. This is depcited in details by the Figure F.2. Therefore, the intermediate approach is suitable for the same reasons explained in Section 4.2.

For what concers the Preparation Order submodule, we adopt an integrated approach because the procedure to set up the picking list from the preparation orders is shared between the two storage areas of the plant, even though orders arriving from each storage area are stored in a specific buffer with its own waiting time and maximum number production orders before printing the picking list.

Based on the line feeding policy, the Preparation Control submodule generates the preparation orders. Current and future situations, identified with the managers, show that elm.leblanc is currently using two line feeding policies, namely, Kanban and JIS. For the sake of brevity we do not report the identified situations here but in Table F.3. Therefore, for the same reason already explained in Section 4.2, we decide to adopt an independent approach for this submodule.

5.2.2.3 Specification of Function-Triggering Events

Based on the SADT diagram (Subsection 5.2.1.3), Table 5.9 provides the main events that can cause the trigger of a function, hence of the related module.

Table 5.9: The main events triggering each function.

Function (Module)	Event
Prepare	Release of preparation order
Load	End of preparation order
Transport	Release of picking order
Produce	Release of manufacturing order

5.2.2.4 Decisions Impacted by Update Events

Any update of some values appearing in inputs or constraints of some decision associated with one or more (sub)module will impact on the decision itself as illustrated in Table 5.10.

Table 5.10: The main events acting on decisions considered in each module.

Update of	Decisions	Involved Module/Function
max number of preparation orders per line cart type in the picking station milkrun capacity line takt time location of supermarkets	Line feeding policy	Prepare
storage support of component number or location of supermarkets Loading/unloading time milkrun capacity cart capacity or number of stages per cart	Inventory Planning and Control	Load
number of milkruns dedicated to the Line the number of supermarkets the milkrun capacity the production sequence	Routing (Supply)	Transport
the customer demand new process technology number of lines dedicated to a product	Capacity Investment	Produce
set of products processing time of tasks equipments	Line Design	

5.2.2.5 Specification of the Events Impacting on Each Information

Events acting on each decision will automatically impact information related to the module. However, we could consider other events related to the business logic.

For instance, in the case of the Produce module, events such as the release of the preparation order will have an impact on the feature Preparation batch size (P.s.8). Also, the event reporting consumption of components in the workstation will impact the Current line-side inventory (Z.α.9).

5.2.2.6 Building the Structure and Format of KPIs

Based on the formula, Table 5.11 summarizes the KPIs of the Prepare module. The first two columns present the list of attributes (outputs by the DSS) required to compute them, (see section x and appexndix Y), while the right most column indicates the frequency of measure, hence of reporting of the concerned KPI.

A meeting with the project team allows to decide how to represent the KPIs, e.g., in the elm.leblanc case, mean values are represented via tables. This representation format is a more suitable option when managers look for the accuracy and ease of data manipulation. On the other hand, percentage values were presented via histograms to facilitate the comparison of values.

Table 5.11: KPIs structure of the prepare module.

Attribute Name	Code ID	KPI Name	Formula	Unit of measure	Frequency of measure
Daily working time	T.g.4	Mean time to prepare one preparation order per storage area	$\frac{1}{R.c.17} \times \sum_{i=1}^{N=T.g.2\times T.s.1} \frac{T.w.7(i)}{N}$	[min]	Daily
# of periods	T.g.2				
# of orders/d	P.θ.1				
# of items stored	Z.g.6				
Qty/order/ item	P.θ.3	Mean time to prepare a list of preparation orders per storage area	$\frac{1}{R.c.17} \times \sum_{i=1}^{N=T.g.2\times T.s.1} \frac{T.w.7(i)}{P.s.8}$	[min]	Daily
# of zones	R.g.8				
# of pickers/zone	R.c.17				
Avg # of bins/ut	T.s.1	per-shift usage % of preparators	$\sum_{i=1}^{Z.g.6} \frac{T.w.7(i)}{T.g.4\times R.c.17\times R.g.8} \cdot \ P.\theta.3 \times \ P.\theta.1$	[%]	Per Shift
Item picking time	T.w.7				
Preparation batch size	P.s.8				

5.2.2.7 Data Structures

Table 5.12 provides the structure of the *data set collection* of the Prepare module. The unique ID indicates the elements involved in identifying the row and column of data or the level of the data tree structure.

Table 5.12: Data structure of data set of the Prepare module.

Data set collection	Elements based on which data are indexed
Preparation time	Process, Storage support
PreparationDataset	Workcenter/Line, Storage area, Date, Shift
Inventory	Component, Workcenter/Line, Supermarket
BOM	Workcenter, Line, Workstation, Product, Component
ComponentFeatures	Component, Workcenter, Line, Workstation, Product
StorageSupportDimensions	Storage support

5.2.2.8 Drawing an Organogram

Eight departments have been involved in the project as specified in Subsection 5.2.1.10. From each department, actors with different skills contribute during the life cycle of

MEMO I4.0. The organogram represents the organization model that must be generated in the design phase in order to associate the responsibility of data management of each module to the actors. For confidentiality reasons, we do not provide the elm.leblanc company organogram.

5.2.2.9 Instantiation of the Metamodels of Views

As indicated in Subsection 4.4.2.9, we use OCL (the Object Constraint Language) to define the constraints to be applied to the UML diagrams already defined. In addition, generalization and multiplicity of all relationships must be specifically instantiated for the elm.leblanc case at study.

In the following, we describe how UML diagrams for each view and related to the Prepare module have been defined. For the sake of conciseness, diagrams and related tables are not reported here but in X. In them, the green color is used for OCL constraints, class names, or multiplicities to indicate the specificity of model in the elm.leblanc case.

- **functional model (of the Prepare module):**

  A functional model of the Prepare module instantiated from the metamodel of the function view (see Figure 3.2) can be derived as follows.

  OCL constraints show that in the scope of the MEMO I4.0 project of elm.leblanc, the storage activity includes only the prepare function (expressed by the size of *Activity.Functions* being set to 1 Therefore, we assume that other functions as e.g., receive or store, do not impact plant configuration. Likewise, we use the OCL constraint of the *Function* class to illustrate the MS process, i.e., the *Prepare* function is the first function in the process, and the *Load* function is the only membre of the set of its *successor*.

- **process model:**

  Similarly, the process model of the Prepare module can be instantiated starting from the metamodel of the process view (see Figure C.2). The resulting diagram is shown in Figure F.2. We use stereotypes for the specialization of different classes according to the elm.leblanc requirements, while Guidelines and DNA (Subsection 5.2.1.2) are used to specify the OCL constraints for the elm.leblanc plants. The studied plant can be considered like a DMS since all lines in the workcenter are dedicated lines, each of which is an element of type *Workcenter/line* in Table 5.7. Furthermore, all lines are *Uline* type, and the *Workstations* are of *Station* type and occupy *Resources* of type *Space*. Finally, one *Product* belongs to only one *Product Family* and has one BOM.

  Despite BPS recommending the use of the pull principle, when analyzing the practices within the studied plant, we remark that the pull principle is only

applied for the long-term production planning, whereas for the short-term production planning the push principle is used instead. This case demonstrates that in addition to considering the company guidelines, it is also essential to consider the specific practices and techniques used in each plant. Nevertheless, we designed the Produce module with a *partial* level of genericity to allow for its use in other elm.leblanc or Bosch plants, so that the module can accommodate pull, push and hybrid production principles.

The preparation order is generated by a workstation to supply one component. Although we initially assumed that there would be no shortages in the storage area, it would be advisable to add a precondition to verify that the quantity of the component in the storage area is sufficient to meet the demand specified in the preparation order. This would avoid prevent the model execution from being blocked due to insufficient stock. The preparation order status can only have one of the three values, namely, "*Created*", "*Release*", and "*End*".

The kanban or JIS is the two-line feeding policy used to manage the component supply. We can store the component either in the main warehouse or the advanced area, depending on whether the storage strategy is shared or dedicated, respectively. Finally, only one storage support is associated with one component. Three types of storage support are possible in the studied plant, *Standard Box*, *Cardboard Box*, or *Pallet*.

- **decision model:**

  Of course, several decisions could be considered in each module. However, together with elm.leblanc managers, we consider decisions frequently occured and those expected to occur often in future scenarios. For example, in the studied plant, a common problem for managers or engineers concerns the decision of which line feeding policy is more efficient for a new or existing component. Therefore, in the Prepare module (as described in Subsection 5.2.1.7), we define the line feeding policy to support them in this decision. Note that the line feeding policy is considered an attribute of the component, hence an input for the simulation model. Therefore, the line feeding policy can be determined either according to a what-if scenario, based on the results of an optimization model, or using actual data.

  Figure F.3 depicts the UML diagram showing the model of the decision view. In it, CIAOC properties (Subsection 3.4.4) are stereotyped as Features. Table F.12 defines the inputs and constraints to be considered in the line feeding policy decision, according to what we discussed and identified in Chapter 3 (see Table B.8). If other inputs or constraints are required to be considered by elm.leblanc to make this decision, then, they will considered as settings since it is

specific to the process of the studied plant (Subsection 5.2.1.11). In our case, only one assumption needs to be made when making this decision, i.e., considering the storage space to be independent of the storage strategy (R.a.24). Likewise, as shown by OCL constraints in the diagram, only one output and one criterion will be considered. Furthermore, since only two line feeding policies are handled in the considered plant, the output values can only be "*Kanban*" or "*Just In Sequence*" *(JIS)*.

- **performance model:**

  Based on the metamodel of the performance view (Figure 3.6), we stereotype the KPI as an attribute.We use OCL constraints to illustrate that in case a KPI is an MS feature, the attribute codeID is a concatenation of Class, Category, and NumberID. Otherwise, when a KPIs is not included in the MS features already defined (as e.g., sustainability-oriented KPIs, see Subsection 2.3.3.5 and Subsection 2.4.1), then, its unique ID is obtained by incrementing the number of KPIs.

  At the definition phase, we identified the KPIs involved in the prepare module (see Table 5.5). OCL constraints are then used to associate each KPI with the corresponding formula, the frequency measure, and the unit of measure specified in the Table 5.11. The resulting diagram is shown in Figure F.4.

- **information model:**

  We instantiate the metamodel of the information view illustrated in Figure 3.5, and we define the OCL constraints using the events defined in Table 5.9.

  The release of the preparation order triggers the prepare module, impacting two features, namely, the flag of launching the preparation order, T.s.2, and the number of preparation orders per line, P.θ.1. Each feature matches one attribute, a boolean attribute corresponding to T.s.2, and an integer attribute for P.θ.1, which must not exceed the maximum number of preparation orders per line before printing the picking list in the storage area. The diagram of Figure F.5 depicts the obtained model.

- **data model:**

  The instantiation of the metamodel data view defined in Figure C.3 allows to derive a model for the data view of the Prepare module, which is shown in Figure F.6. During the realization of the MEMO I4.0 project, elm.leblanc was undergoing a migration from the "*MAC-PAC*" ERP to the "*SAP*" ERP. Therefore, the two systems coexisted, and each one was the owner of a part of the dataset. Furthermore, other datasets were handled some spreadsheet-based tools.

The three EIS are used for providing the necessary datasets required to generate the function data related to the prepare module defined in Table F.12, namely: the *PreparatorsData*, the *preparaionParameters*, the *InventoryData*, and the *ComponentsData*. For instance, the *preparatorsData* are used in the Prepare and Load modules. Therefore, they correspond to both the prepare and the load functions. We exploit two datasets to generate the *PreparationParameters*. The first is the preparation time provided by the start and end messages of the preparation via the TCP-IP protocol. The second corresponds to the *PreparationDataset* extracted from the spreadsheet system. The *ComponentsData* are generated by three datasets, the *BOM* provided from the *SAP* ERP in JSON data format, the *ComponentFeatures* and *StorageSupportDimensions* extracted both from the spreadsheet in CSV data format. Finally, the *inventorydata* are generated by the *Inventory* dataset provided by the *MAC-PAC* ERP in CSV data format.

- **resource & organization model:**

  By instantiating the resource & organization metamodel defined in Figure C.4, we can derive a model for the corresponding view of the Prepare module: however in the studied plant, the concept of resource capability and organizational unit are not used and can be discarded. The resulting model is depicted in Figure F.7.

  The preparator is stereotyped as a resource dedicated to only one storage area. It is an actor (logistic technician) that belongs to the logistics department and plays one role (picker), this latter requiring some skills. Preparators of the same shift are assigned to different storage areas. Moreover, one preparator can be assigned to different shifts on different dates. The shift is decomposed into four time slots: the first corresponds to the first ten minutes when resources are present but not available for task execution, the second and last time slots correspond to the working time, separated by a break.

5.2.3 Implementation Phase: Verification & Validation of Model Behavior of Each Module

The main goal of this phase is to build both the simulation and optimization models, check their behavior, and validate them w.r.t. real-world results. We follow the steps defined in the Chapter 4 to achieve this, and for each module.

We begin by choosing the technical solutions to capture the raw data, before creating the DB tables corresponding to the data model and building the layers of the development architecture (data, business, mapping, and presentation). We recommend generating the pseudo-code describing the module behavior before building the simulation model. Finally, we evaluate the genericity and flexibility of the module which is the object

of the undergoing verficication by analyzing traceability files, conducting experiments, performing corner tests, and then displaying relevant KPIs. The following sections detail these steps.

5.2.3.1 Choice of the Technical Solution to Capture the Raw Data

Several technical solutions must be put in place to ensure vertical integration from devices to EIS. As we deal with a brownfield scenario, constraints exist in the choice of new technical solutions, as they must fit with previously existing ones. In our case, the first problem encountered is that not all workstations can be connected to the Edge or Cloud. As a consequence, for the concerned workstations keeping track of the start|end time of the processing of a task may become hard to achieve.

This issue is solved by using the start|end time of connected workstations preceding the non connected workstation. MES allows real-time data from connected workstations. We use resources (e.g., operators) to obtain the actual data of the studied plant, such as the start and end of tasks. In addition, some workstations have been equipped with smart resources (e.g., exoskeleton) to obtain ergonomic data related to the operation execution by an operator. Below, Table 5.13 provides a list of some trigger messages to retrieve in order to compute information that will be used in the simulation model of the modules.

Table 5.13: Technical solutions to capture data raws.

Data raws	Technical solution	Information
Start|end time of preparation	Preparator	Preparation time
Start|end time of loading	Preparator	Loading time
Start|end time of unloading	Transporter	Unloading time
Start|end time of operation	Operator / Barcode reader / Robot	Processing time

5.2.3.2 Creation of the Database Tables Required by the Data Model

As indicated in Subsection 4.4.3.2, in this step, we need to decide on the database management system. In the case of elm.leblanc, we chose the Microsoft SQL Server Management Studio 18 since it is already used with several other applications, making data sharing much easier. We created a database called "*SmartFactoryDB*", adding all the tables corresponding to the modules. Each time a new module is required, we identify and create the new tables needed. Figure 5.4 shows a screenshot of the "*SmartFactoryDB*" database.

5.2.3.3 Creation of the Data Access Layer

The main objective of the data access layer is to establish a connection with the database and manage interactions with it. To achieve this, we create a Data Access Object (DAO).

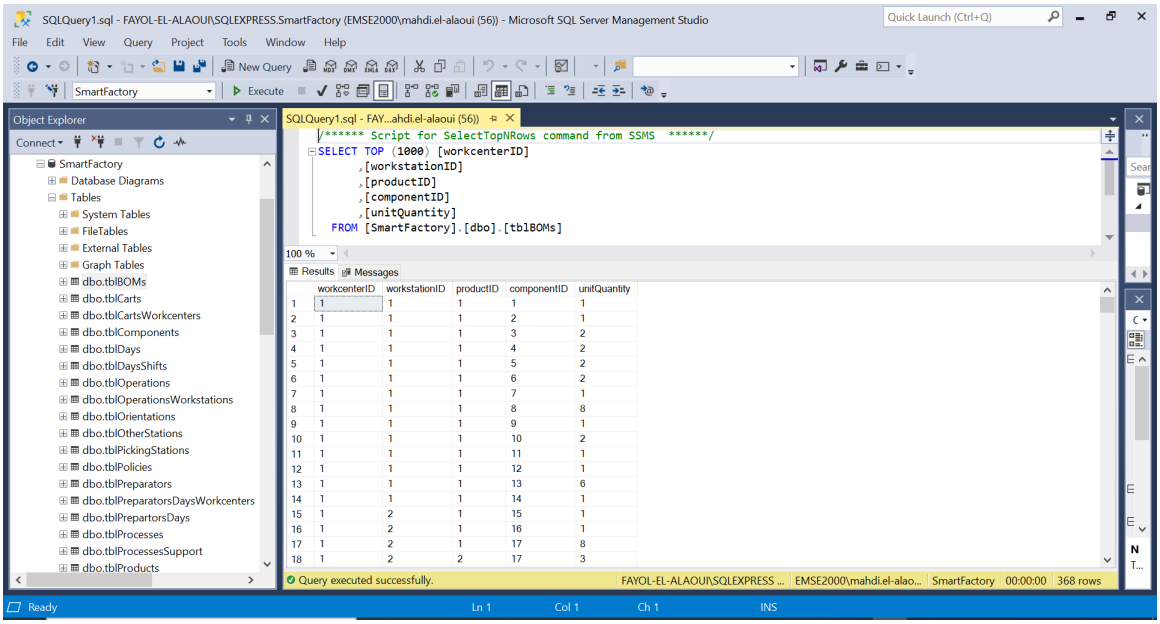

Figure 5.4: a screenshot of the "*SmartFactoryDB*" database.

This DAO allows to have a relationship between the program and the database, enabling to perform essential operations, namely, Create, Read, Update, and Delete (CRUD) on different tables.

Traditionally, each table would have its specific DAO. For example, if we have a table called "Preparator", we would create a corresponding PreparatorDAO to handle database interactions for that table.

However, as indicated in Subsection 4.4.3.3, we decided to take a different approach. Instead of having one DAO for each table, we created a DAO for each function. By doing so, we can directly instantiate the specific module using its associated DAO. Consequently, we end up with DAOs for Prepare, Load, Transport, and Produce functions, simplifying working with each module independently.

We often combine data from different database tables when setting up a module. When the combined outcome is valuable for business needs, we create data views in the database to showcase the results. We rely on procedures and scalar functions to export these results directly to external systems like spreadsheets.

Screenshots of procedures and scalar functions used to instantiate the modules are given in Figure F.8. Additionally, Figure G.1 presents a function example demonstrating how DAO can leverage queries using a database-defined procedure.

5.2.3.4 Building the Business Model Layer

The business layer is independent of the modeling platform, and linked to the business logic of the company instead, using the data access layer to build the model corresponding

to the considered module. For instance, the business layer of the Prepare module is composed of the Function data described in Table F.12. PrepareDAO allows to read the PreparationParameters, PreparatorsData, InventoryData, and the ComponentsData, then builds the business object corresponding to the Prepare module. Similarly, to create the business objects corresponding to the Load, Transport, and Produce modules, we use the function data of each module defined in the Tables F.13 to F.15, respectively.

5.2.3.5 Creation of the Mapping Layer

The mapping layer must take into account the architecture of the chosen or available platform. In our case, we chose Arena software as a platform for simulation. Based on our analysis of the elements of Arena software, we implement simulation objects from each module object. Figures G.3 to G.16 provide a list of class and functions to transform a module object to a simulation object. These functions can be reused in other simulation projects that utilize the Arena Software.

5.2.3.6 Writing of the Pseudo-Code of Modules

The pseudo-code describes the function in terms of products, processes, resources, and rules that must be respected. The Prepare module offers the possibility to see how the pseudo-code of modules is writed.

As already stated, relasing the Preparation Order triggers the Prepare function. The pseudo-code of the Prepare module Algorithm 1 allows the generation of preparation orders in a specific workcenter (WC), for each line and workstation. First, we set the variable flag of launching the Preparation Order of each component to zero, i.e., we start with the condition that no orders have been launched. Then, we call the procedure of generating the preparation orders, which the read can find in Appendix E (Algorithm 6).

Algorithm 1 Pseudo-Code of Prepare Function.

Input: WC **Output:** $PRO\ status$

Require: $New\ preparation\ order$

Ensure: $All\ PRO\ status\ are\ End$

```
NL = getNumberOfLines(WC)
for l ← 1, NL do
    NWS = getNumberOfWorkstation(l)
    for w ← 1, NWS do
        n ← getSizeOfComponentListOfWorkstation(w)
        for k ← 1, n do
            FlagOfLaunchingPRO(C_lwk) ← 0
        end for
     GeneratePreparationOrdersFromWorsktations(l, w)
    end for
end for
```

5.2.3.7 Building and Solving an Optimization Model, and Conducting the Numerical Experiments

As we said in Section 5.2, in our case studies, two approaches will be considered when using MEMO I4.0 to develop a DSS: using simulation alone or combining optimization and simulation. If a DSS based on simulation optimization is to be derived, additionally, the following steps must be performed in the implementation phase:

- building and solving of the optimization model.
- choosing the solving approach, such as heuristics or exact methods.
- mapping the decision variables of the optimization model with the simulation model attributes.
- conducting numerical experiments.

Among the three case studies presented in Chapter 5, we present the details of this step in the third case study, in which we deal with the line design problem involved in the Produce module (c.f Appendix F.6.2).

5.2.3.8 Building a Simulation Model of Modules

The creation of the simulation model of the module is based on four elements, which, in the case of the Prepare module that served as an example, are:

- The modeling choices defined in the Table 5.8.
- The pseudo-code corresponding to the module (e.g., Algorithm 1).
- The data and associated structures that will describe the simulation object (e.g., entity attributes, local and global variables, sets, see Tables 5.12 and F.17).
- The KPIs that must be computed (see for example, Tables 5.5 and 5.11).

Since MEMO I4.0 leads to data-driven models, all simulation objects used for modeling a module can be customized through data while respecting the scale level, i.e., by defining the attributes of the simulation object that allow it to be specified at the scale level of the module in which he is involved.

Among the function data, some attributes will play the role of inputs and outputs. Generally, the choice is made with experts so as to select the variables that most significantly impact the process. For the Prepare module, Table 5.14 summarizes the input and the output variables that have been chosen in the case of study.

Table 5.14: Variables of design of experiments

Inputs	**Designation**
X1	Waiting time before printing preparation order
X2	Preparation time of standard box
X3	Preparation time of cardboard box
X4	Preparation time of pallet
X5	Takt time
Outputs	**Designation**
Y1	Mean time to prepare the preparation order
Y2	Mean time to prepare the picking list
Y3	Per-shift % usage of preparators

We use historical data to identify the distribution that better fit each input variable: the data correspond to 3 weeks of production. Then, we resort to Anderson-Darling statistical hypothesis test to check whether the assumption of the normal distribution of the variables is correct (Anderson, 2011). Figure 5.5 shows the probability plot of the variables X1, X2, X3, X4 and X5.

Since the p-value is less than the significance level (alpha=5%) for the variables X1, X3, and X5, we reject the null hypothesis, and consider that they do not fit the normal distribution. we keep normality assumptions for $X2 \sim \mathcal{N}(\mu = 139.2, \sigma^2 = 94.2452)$ and $X4 \sim \mathcal{N}(200.3, \sigma^2 = 22.4676)$. Note that several methods could be used to estimate the parameters of a distribution. Here, we use the Maximum Likelihood Estimation (MLE).

For variables X1, X3, and X5, we use the Python library (SciPy) to fit a probability distribution to each variable (Virtanen et al., 2020). We estimate the parameters for 40 distributions. Then, we use the Kolmogorov-Smirnov(K-S) test to assess the goodness of fit (Godina et al., 2017). Table F.18 provides a full report of the results of the test.

Figure G.2 gives the p-value of the K-S test for different α, and it is used to interpret the results. Table 5.15 reports the computed p-values of X1, X3, X5 and shows the detected distributions and associated features.

Table 5.15: P-values of K-S test for X1, X3, X5 variables of the Prepare module.

Xi	**N**	**P-value of K-S test for $\alpha = 5\%$**	**Distribution**	**Statistics**
X1	1442	0.03581	Triangular	0.01953
X3	260	0.08434	Halfcauchy	0.19626
X5	1442	0.03581	Uniform	0.02041

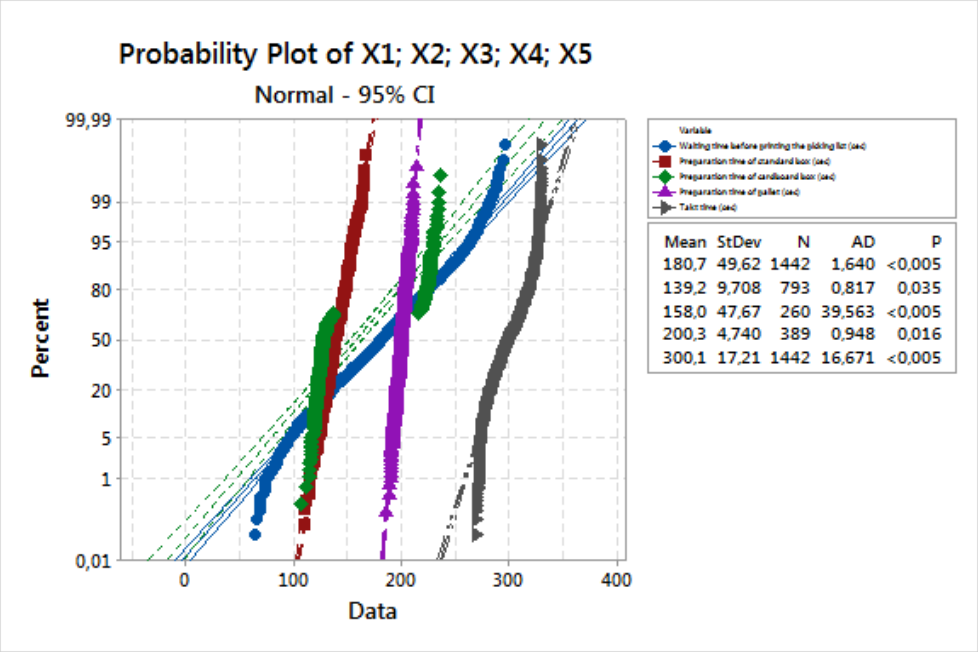

Figure 5.5: The probability plot of the input variables impacting the Prepare module.

Based on these results, we can say with 95% confidence that $X1 \sim Triangular(a = 0.52, b = 59.63, c = 240.10)$ and $X5 \sim \mathcal{U}(a = 270, b = 60)$. However, we reject the null hypothesis for X3 since the p-value is less than the distribution statistics.

In order to understand the behavior of variable X3, we further analyzed the process. We noticed that, unlike We made a first remark about the storage support. In contrast to the standard box and the pallet, used only in storage areas 1 and 2 respectively, the cardboard box support is shared between the two areas. Thus, we deduced that the mean preparation time for the cardboard box support must be specified according to the storage area. We then decomposed the variable X3 into two variables, X3-1, and X3-2, to indicate the preparation time of the cardboard box support in storage areas 1 and 2, respectively. Then, we repeated the distribution fitting procedure, and obtained the results of the Table 5.16. Based on these results, we accept the null hypothesis, and conclude that $X3-1 \sim logistic(\mu = 124.74, s = 2.87)$ and $X3-2 \sim Beta(\alpha = 2.03, \beta = 2.46, p = 215.18, q = 22.50)$

Table 5.16: P-Values of K-S for X3-1, and X3-2 variables indicating the preparation time of cardbbox support in storage area 1, and 2, respectively.

X3-i	**N**	**P-value of K-S table for $\alpha = 5\%$**	**Distribution**	**Statistics**
X3-1	175	0.10280	Logistic	0.05908
X3-2	87	0.14580	Beta	0.07459

5.2.3.9 Creation of Traceability Files

Table 5.17 provides the attributes list that identify the traceability files of each module.

Table 5.17: Attributes of traceability file of each module.

Prepare	**Load**	**Transport**	**Produce**
Replication	Replication	Replication	Replication
Day	Day	Day	Day
Shift	Shift	Shift	Shift
Workcenter	Storage Area	Picking station	Launch Day
Line	Picking station	Loop	Workcenter
Station	Loop	Line	Line
Component reference	Line	Transporter	Product
Inventory	Preparator	Number of standard cart picked	Production resource
Storage area	Number of boxes per cart stage	Number of LKE cart picked	Produced quantity
Supply type	Number of standard carts filled	Time of pick up	Inventory
Launch time	Number of the LKE carts filled	Supermarket	Blocking Time
Print time	Start loading time	Number of boxes unloaded	Start production time
Preparator	End loading time	Number of empty boxes loaded	End production time
Size list PRO	-	Delivery Time	-
Start preparation time	-	Start unloading time	-
End preparation time	-	End unloading time	-

5.2.3.10 Launching the Experimentations

Before launching the experimentations, we analyze the output values (Y1, Y2, Y3) of the Prepare module collected from the historical data of the real system. Then, we use the box plot to quickly identify the outliers and the dispersion of the data set.

Based on the results of this analysis (which the reader can find in Figure F.9), we find that the variable Y1, i.e., the mean time of the preparation order in storage area 2 (the main warehouse) is more significant than in storage area 1 (the advanced area). On the one hand, in the advanced area, the picker uses the Kardex to retrieve the standard boxes or transports the cardboard boxes already stored in the area. On the other hand, in the main warehouse, the picker uses the forklift within the aisles to retrieve the pallets and the cardboard boxes, and it takes more time to manipulate the MHE.

The second box plot of Figure F.9 illustrates the same conclusion as the first box plot of Figure F.9, i.e., the mean time to prepare the picking list in the main warehouse is more significant than in the advanced area. Although the storage of components in the main warehouse of the studied plant follows the shared strategy, we remarked that a practice exists that consists in organizing the storage according to the ABC classification. If, in the picking lists, the preparation orders concerns component of class A and component

of class B or C, the consequence is that the routing for the picking will take more time than storing these components in the Kardex and retrieving them.

The last box plot of Figure F.9 shows that the variable Y3, the usage of preparators is, in the advanced area more important than in the main warehouse. This difference can be explained by the number of preparation orders generated from workstations in each area. During the three weeks considered for the historical data, there had been 967 preparation orders for the advanced area, and only 475 preparation orders for the main warehouse.

We use the same initial conditions as the real system and retrieve the initial components inventory at supermarkets, then we simulate three weeks (16 days) with two shifts per day, using the distributions that we have found for the variables X1, X2, X3, X4, and X5. The comparison of simulation results of 300 replications with the historical data is given in the Table 5.18.

Table 5.18: Result of the simulation tests of the Prepare module and comparison with historical data.

	Historical data of 3 weeks (1442 Preparation orders)		**Simulations results for 300 Replications (minutes)**		
Y1	*Actual Mean time to preprare the preparation order (minutes)*	*Standard deviation*	*Mean time to prepare one preparation order*	*Standard deviation*	Gap
Storage Area 1 (967)	7.56	0.89	7.86	0.57	3.82%
Storage Area 2 (475)	9.84	0.90	9.44	0.64	-4.24%
	Historical data of 3 weeks (320 Picking list)		**Simulations results for 300 Replications (minutes)**		
Y2	*Actual Mean time to preprare the picking list (minutes)*	*Standard deviation*	*Mean time to prepare the picking list*	*Standard deviation*	Gap
Storage Area 1 (190)	26.85	8.41	27.32	5.26	1.72%
Storage Area 2 (130)	2873	8.91	28.27	4.87	-1.63%
	Historical data of 3 weeks (2 shifts/day)		**Simulations results for 300 Replications (minutes)**		
Y3	*Estimated per-shift usage |% of preparators*	*Standard deviation*	*per-shift usage |% of preparators*	*Standard deviation*	Gap
Storage Area 1 (32)	38.88%	6.30%	40.58%	1.72%	4.19%
Storage Area 2 (32)	28.47%	5.60%	28.53%	1.03%	0.21%

5.2.3.11 Running the Corner Tests

We present in the following the sensitivity analysis performed for the Prepare module. We decide to illustrate only the two cases of X1 and X2.

Sensitivity analysis of variable X1:

We consider for X1 (waiting time before printing the preparation order) values from 0 to 220 seconds, while X2, X3-1, X3-2, X4, and X5 are given constant values. The tests show that the time of preparation of the preparation order (Y1) is not affected by the waiting time, while a slight impact on the preparation of the picking list (Y2) in the storage areas can be observed when increasing the waiting time X1. Likewise, we remark that per-shift usage by preparators (Y3) decreases slightly when X1 increases, which is to be expected because the number of preparation orders will also be impacted once the printing time increases. Figure F.10 shows the plots of Y1, Y2, and Y3.

Sensitivity analysis of variable X2:

Similarly, we plot Y1, Y2, and Y3 values when the value of X2 (preparation time of the standard box) is varied from 130 to 149 seconds, while X1, X3-1, X3-2, X4, and X5

are given constant values. Figure F.11 shows the result. Since the standard box is stored only in area 1 (AFLI), we present only the output values dedicated to the storage area 1. We remark that Y1 increases when X2 increases. We explain this effect by the effort of the operators to catch up on waiting time. In comparison, Y2 is only slightly impacted, probably due to the picking list size being very similar. Finally, as X2 increases, Y3 also increases, which is to be expected since more time is spent preparing the standard box, thus, the operators are busier.

5.2.3.12 Displaying the KPIs

In the elm.leblanc case, with the project team, we decided to display KPIs by means of a table to represent mean values of the Prepare, Load, and Transport modules, while we chose a histogram tool to represent the percentage values.

As we said in Subsection 4.4.3.15, the Performance Management System (PMS) displays these KPIs for the managers.

5.2.3.13 Checking the Module Agility and Flexibility

For the prepare module, we fixed the genericity degree at *specific*. We now need to check if the model obtained can represent all possible cases that must be covered.

In order to do so, we tested some scenarios which the managers and experts consider as capable of stressing the performance of the module:

- Two lines in the workcenter.
- Each line with three workstations.
- Line with mixed product (3 products).
- Common components between products.
- Kanban policy with configurable triggering quantity and delivered quantity.
- JIS policy with configurable offset time of launching production order and delivered quantity.
- Different preparation options, i.e., shared or not, of the preparators between lines.
- Multiple strategies for selecting the preparator responsible for preparing the picking list, such as a static approach that does not consider workload balance or a dynamic approach that distributes tasks evenly among preparators.
- Preparation time depending on the storage support.

Likewise, we define test scenarios to check the flexibility of the Prepare module regarding the scale level that we had chosed at *product level*. For instance, for one component reference required in different workstations, be they in the same line or not, we test if we can handle the following cases:

- Two line feeding policy (Kanban and JIS).
- Two storage area (Main warehouse and Advanced area).
- Different supermarket capacity.
- Three storage supports (Standard box, Pallet, and cardboard box).

5.2.4 Maintenance Phase: Versioning of the Model of Each Module and Generating the Corresponding Documentation

As explained in Subsection 4.4.4, in the maintenance phase of the development cycle, we need to generate or update the documentation associated with the models of each module and maintain their versioning. In order to do so, we follow a continuous improvement approach when developing the required functionalities for each module. It entails developing a module version that initially incorporates only a subset of functionalities and progressively integrating additional functionalities in subsequent versions until all the necessary functionalities are included.

Of course, following the evolution of the guidelines and best practices of elm.leblanc, and based on the manager/director requests, an update of the existing functionalities may become necessary, leading to add functionalities not planned at the beginning. These evolutions will lead to new versions of the module in the library.

5.2.4.1 Documentation and Versioning of Module Models

A documentation must be generated for each module. In what follows, we provide an example of documentation generated for elements involved in the Prepare module (see Table 5.8).

For the versioning of different models, we use git to monitor the development and tests of modules.

5.2.4.2 Checking the Compatibility Among View Models of Modules

As mentioned in Subsection 4.4.4.2, the incompatibility between the requirements of one view model of a module and another is checked based on the data model and the OCL constraints.

In our case, most elements use the number of lines, workstations, products, supermarkets, and shifts as parameters to generate the data set required for the simulation model. We group these data sets in general data (see Table F.16) and consider them as a *"CommonObject"* i.e., a common ancestor s.t. all function data of modules can inherit from it. Therefore, we ensure compatibility among views even when integrating modules.

5.2.4.3 Organization of Review Meetings with the Project Team

In all the review meetings with the elm.leblanc team animations when during the simulation are used to help checking the correctness of the model behavior regarding the real system. Later, the obtained KPIs were to discuss and decide whether the module could be validated.

Table 5.19 summarizes the number of versions currently developed for each module and the number of review meetings that took place. Each version must be validated in the meeting before developing a new module version.

Table 5.19: Module versions and the number of meeting reviews per module.

Module	Number of versions	Number of meeting reviews
Prepare	12	8
Load	16	10
Transport	25	18
Produce	23	15
Integrated modules	16	13

5.2.4.4 Updating Documentation and Versioning of Models of Modules

Several versions of the simulation model of the Prepare module have been created as shown by Table 5.20.

It is not recommended to build a model including all the expressed needs at once. Rather, we suggest adopting a continuous improvement approach to enhance module functionalities and quickly obtain the most advanced version, which involves refining the module and validating its performance after each improvement step.

5.3 Operational Stage: Investigating of the Internal Logistic by Simulation

In this use case, the request from the managers was to assess the performance of the internal logistic, as well as to understand how the inventory of some components evolves

Table 5.20: All versions of the simulation model of the Prepare module.

Module	Version	Main Added Functionality	Status
Prepare	V-01	Model the generation of preparation orders and the preparation process	OK
	V-02	Consider only kanban policy	OK
	V-03	Control the preparation orders, Inventory, resources and shifts	OK
	V-04	Assume preparation time independent from component storage support	OK
	V-05	Use of real data providing from plant and considering a simplified version of JIS policy	OK
	V-06	Take into account that the preparation time is dependent of component storage support	OK
	V-07	Consider the preparation order with line, workstation, product	OK
	V-08	Include shared preparators between lines	OK
	V-09	Model BOM as advanced set and consider common components between products	OK
	V-10	Add component pictures	OK
	V-11	Improve the traceability files for tracking the preparation process	OK
	V-12	Include an improved version of JIS as supply policy	OK

during the production cycle if the line feeding policy is changed.

In the following, we explain each phase of the operational stage. After analyzing the needs of managers in the definition phase, we go back to the strategical/tactical stage for developing the required modules.

5.3.1 Definition Phase

In order to focus on the internal logistic, we decide to include only two activities, namely, supply and storage. The production activity is considered like a black box, producing pieces according to the takt time of the line.

Since before this first use case proposed by elm.leblanc no module existed yet (see Figure 5.1), the first decision concerned which modules needed to be developed to give the managers accurate answers. Therefore, we decide to focus first at all on the preparation function of the storage activity, based on the experience of managers, who have observed that picking and preparing boxes/carts can be time-consuming, and their willingness to decouple the reception and storage flow from the picking flow. However, we consider that capturing the loading and transport functions of the supply activity is necessary as well.

This is why we started by developing three new modules: Prepare, Load, and Transport, corresponding to the three mentioned functions. We did it in the development phase (Section 5.2) by following the roadmap described in Figure 4.10 for compacted version or in Appendix E (Figure E.1) for detailed version.

5.3.2 Design Phase

As soon as the required modules (Prepare, Load, and Transport) are inserted in the library of modules, we integrate them and instantiate their models according to the requirements of managers.

In the following we shortly describe the instantiation of the conceptual models of the different views of the three modules.

In the instantiation of the functional model, we consider one function (prepare) of the storage activity, and add two functions (load and transport) of the supply activity, the reader can refer to Figure F.18. The relationships among functions give precedence constraints in the MS process: for example, the prepare function precedes the load function, which precedes the transport function. This instantiation is possible because it is compatible with the OCL constraints expressed in the Figure F.3.

For what concerns, the instantiation of the decision model, the object diagram of Figure F.19 illustrates the case of the studied elm.leblanc plant. Four assembly lines are represented. We only show the product objects of one of them and give some examples of objects of their bill of materials. Likewise, we provide an example of the preparation order "PRO1" generated for the component "Bruleur WB6", which is stored in the warehouse area in a support of type pallet and managed with the JIS policy.

An example of instantiation of the performance model is given in Figure F.21: in it, the objects illustrating the KPI 1,2 and 3 incorporate the same information represented in the Table 5.12, as these KPIs have been decided in the development cycle. If other new KPIs are required in the execution cycle, we need to update the module, by going back to the development cycle.

As we can see in the models represented by the object diagrams, we fix the parameters of each element (e.g., the functions belonging to each activity and their types, the number of Workcenters/Lines and their types, the number of products and their production policy). The goal is to have the settings of the MS configuration we aim to simulate.

5.3.3 Implementation Phase

In this phase, according to the development architecture proposed in Figure 4.17, we can generate the simulation objects from the business models defined in the design phase.

We tune the instance models to obtain a simulation model of the internal logistic process we are interested in. We change the line feeding policy of the three components (87186846992, 87167621154, 87186602275) from the Kanban policy to the JIS policy. We simulate one day of two shifts (450 min/shift) and consider ten replications of our simulation model.

Table 5.21, Table 5.22 are examples of two among the studied KPIs. The KPIs results obtained are assessed in review meetings. They are both consistent and adhere to real-world conditions, thus stand validated.

Simulation results indicate that the percentage usage of preparators in the first shift is higher than in the second shift in both storage areas. Our interpretation of this result is that the quantity delivered of some components in the first shift partially or completely

covers the needs of the second shift. Therefore, fewer preparation orders are generated for the preparators of the second shift. Likewise, we remark that the preparators of the same shift are in percentage more requested in the AFLI area than in the warehouse area, which is to be expected since around 60% of the components are stored in the AFLI area.

	% Usage of preparators	
	Shift 1	**Shift 2**
AFLI Area	62.08	41.58
Warehouse Area	51.82	28.54

Table 5.21: Per-shift percentage usage of preparators in each storage area.

The second KPI we report here concerns the trips performed by the transporters. We have fewer loops in the second shift than in the first, which is consistent with the result of the percentage usage of the preparators. However, the average loop duration in the second shift is slightly longer than that of the first shift. This result is probably due to the number of stops to feed the line which is bigger in the second shift despite the number of boxes/carts being smaller.

	Transporters	
	Shift 1	**Shift 2**
Number of loop	18	16
Average loop duration (min)	18.2614	18.8809
Minimum loop duration (min)	15.1712	16.7869
Maximum loop duration (min)	21.9808	21.2438

Table 5.22: Number of loops per shift and durations.

Figure 5.6 provides another example of KPI that we were able to analyze after running simulations. In it, we represent the evolution of the inventory of three components located at the supermarkets of one of the lines of the studied plant during a production cycle (900 min). It can be noticed that the inventory level changes only in two cases: it increases when feeding the line and decreases when producing pieces. The component "87186846992" is an example of a component stored with high quantities like screws: it is an example of a component that after being prepared in the first shift could also cover the needs of the second shift.

5.3.4 Maintenance Phase

In this phase, the utmost priority is enabling knowledge capitalization and module enhancing version improvement.

Concerning the current situation of the internal logistic of the studied elm.leblanc plant, we remark that changing the line feeding policy of the three components from the Kanban

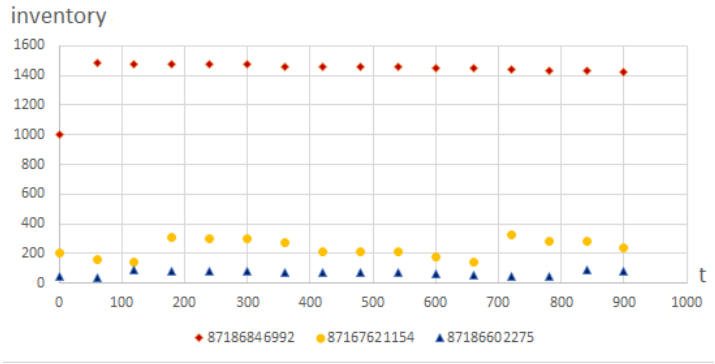

Figure 5.6: Component inventory in supermarkets.

to JIS policy increased the percentage usage of the preparators in the first shift but had no effect for the second shift (see Table 5.18).

Based on the percentage usage of preparators in both shifts, changing the line feeding policies for three components will increase preparator usage by 10%. However, the advantage of using these policies is to free space in supermarkets and provide flexibility to introduce other product varieties that require storing new components. On the other hand, we created a version of the integrated model involving the three modules (Prepare, Load, Transport), and we save the simulation results in database for further analysis.

Therefore, through the implementation of the line feeding policies solutions, the case study has successfully achieved its objectives, which were centered around enhancing the internal logistics of the plant and optimizing stock levels (see Subsection 5.2.1.1) The other two use cases presented in Appendices F.5 and F.6 will address the remaining objectives, namely, meeting customer demand, understanding the impact of the parameters on the plant configuration, managing diverse finished products, and enhancing equipment efficiency.

5.4 Conclusion

In this chapter, we applied the proposed methodology MEMO I4.0 in the real industrial environment offered by elm.leblanc, a French company. A case study has been investigated in which the application of MEMO I4.0 has allowed to design and implement a DSS capable to fit the managers request and improve the internal logistic of an elm.leblanc plant.

In this use case, a simulation approach is applied to assess the internal logistic performance and deal with the capacity investment problem to face the uncertain demand.

We succeeded in providing valuable insights to managers by offering quick responses.

When we started developing MEMO I4.0, among the main goals was to decrease the development time of DSSs tailored to the Industry 4.0 environment. We use the three case studies to reference the time spent developing the modules. The first case study presented in this chapter involving the modules (Prepare, Load, Transport) took nearly six months to complete. Subsequently, for the second case study, developing the Produce module and integrating it with the available modules in the library only required one month. Lastly, updating the Produce module by adding a new KPI and creating a simulation-optimization DSS, as seen in the third case study, was accomplished in just one day. These results demonstrate the efficiency of the capitalization mechanism proposed by MEMO I4.0 in considerably reducing the development time when new manager requests are triggered, and proves its potential in an uncertain environment where agility is the key to industrial effectiveness.

Conclusion & Perspectives (french version)

En conclusion, ce manuscrit commence par identifier les défis et les leviers de la transformation des systèmes de production traditionnels vers l'Industrie 4.0, ainsi que les changements résultants dans le processus de prise de décision. Ensuite, il résume cette transformation en trois évolutions : technique, organisationnelle et transversale.

L'évolution technique concerne les changements dans les systèmes de production dans le contexte de l'Industrie 4.0, y compris la standardisation des échanges de données et les nouvelles techniques et algorithmes. La standardisation des échanges de données est nécessaire pour assurer la cohérence et l'interopérabilité dans les échanges de données entre les composants. Alors que les nouvelles techniques et algorithmes, tels que le big data et l'apprentissage automatique, sont nécessaires pour traiter les grandes quantités de données générées par les nouvelles technologies et améliorer la prise de décision.

L'évolution organisationnelle concerne l'émergence de nouveaux modèles économiques, tels que la tarification dynamique et le paiement à l'utilisation, ainsi que la nécessité pour les entreprises de s'adapter et d'optimiser leurs systèmes de production pour être agiles et efficients afin de répondre aux exigences des clients. Le concept de "leagility", qui combine la flexibilité et l'efficacité, est introduit comme moyen pour les entreprises d'adopter ce mode organisationnel. De plus, il est nécessaire de développer de nouvelles compétences et activités d'emploi en raison de l'émergence de nouvelles technologies dans l'industrie 4.0 et de l'importance de renforcer les compétences des employés pour s'adapter à ces changements. Face à l'incertitude imposée par le monde VUCA, les entreprises doivent s'orienter vers des systèmes de production de plus en plus adaptables. La stratégie "leagility", le système de production centré sur l'humain, ainsi que les systèmes de production adaptables sont présentés comme trois évolutions organisationnelles qui abordent respectivement l'évolution des modèles économiques, le développement de nouvelles compétences, et l'incertitude dans les décisions prises dans le contexte de l'industrie 4.0. La stratégie "leagility" implique la mise en œuvre de stratégies de report et l'utilisation de principes d'efficacité pour les produits standard et de principes d'agilité pour les produits personnalisés, tandis que le système de production centré sur l'humain se concentre sur la place de l'humain au cœur du système de production et sur l'amélioration de l'environnement de travail et des

programmes de formation pour les employés. Enfin, les systèmes de production adaptables permettent d'apporter une nouvelle capabilité d'auto-adaptabilité et auto-configuration et une autonomie de décision leur permettant de faire face aux aléas.

L'évolution transversale concerne le développement des nouveaux indicateurs clés de performances, la surveillance en temps réel, et le contrôle décentralisé. D'abord, le concept d'indicateurs clés de performance (KPI) est mis en évidence, ainsi que leur rôle dans la compréhension et l'amélioration des performances des systèmes de production. Puis, le concept de système de mesure de la performance (Psystème de production) est introduit, il s'agit d'un ensemble de métriques utilisées pour quantifier l'efficacité et l'efficience des systèmes de production. Les KPI classiques dans les système de productions sont divisés en coût, temps et qualité. Cependant, dans le contexte de l'industrie 4.0, de nouveaux KPI, appelés KPI 4.0, sont nécessaires pour évaluer les performances des système de production en raison des changements de modélisation et de comportement, de l'évolution des horizons temporels des problèmes de décision et de leurs interactions. Ces nouveaux KPI sont divisés en Indicateurs orientés durabilité et un indicateur d'agilité. Il est donc essentiel que le développement d'un système de production au sein de l'industrie 4.0 intègre plusieurs perspectives lors de la définition des KPI et soit capable de gérer la complexité et la dynamique de l'industrie 4.0. La surveillance en temps réel implique l'utilisation d'objets intelligents capables de se surveiller eux-mêmes et de demander une maintenance ou une assistance en cas de besoin, améliorant ainsi l'efficacité, prolongeant la durée de vie des composants et facilitant la prise de décision. L'approche de contrôle décentralisé implique la distribution des opérations sur différents nœuds pour faire face à la complexité d'une usine intelligente et permettre l'auto-organisation et la flexibilité.

Ensuite, le manuscrit propose une taxonomie des problèmes de décision dans les systèmes de production et de leurs interactions, ainsi que l'évolution de la prise de décision dans le contexte de l'Industrie 4.0. Grâce à la procédure en trois étapes proposée, nous avons pu structurer et cartographier les problèmes de décision dans les systèmes de production. La première étape consiste à identifier les problèmes de décision dans les systèmes de production, qui sont classés en trois catégories : production, approvisionnement et stockage. La deuxième étape consiste à définir les problèmes de décision en fonction des propriétés: Contraintes, Entrées, Hypothèses, Sorties, Critères (CIAOC). Enfin, la troisième étape consiste à étudier les interactions entre les problèmes de décision, qui sont réalisées par la causalité et les boucles de rétroaction. Une cartographie des problèmes de décision est proposée, qui représente les relations et les dépendances entre les problèmes de décision identifiés. Cela offre un point de départ aux chercheurs et aux praticiens pour considérer les problèmes de décision, leurs interactions dans les systèmes de production et les changements dans la prise de décision au sein de l'Industrie 4.0 et ce qui suit.

Sur la base de notre analyse approfondie de la transformation des systèmes de

production dans le cadre de l'Industrie 4.0, en mettant principalement l'accent sur la prise de décision, nous avons proposé une nouvelle méthodologie de modélisation d'entreprise, appelée MEMO I4.0, pour remédier au manque d'une approche pouvant soutenir la modélisation et la mise en œuvre de systèmes d'aide à la décision orientés Industrie 4.0. La méthodologie proposée a été appliquée pour dériver un système de prise de décision moderne adapté à l'Industrie 4.0 à travers des études de cas, ce qui a démontré son efficacité et son efficience dans ce contexte.

MEMO I4.0 est une approche axée sur les données qui vise à créer un jumeau numérique d'un système de production qui soit évolutif, reconfigurable et utilise des données quasi temps réel. L'objectif ultime de MEMO I4.0 est d'offrir une méthodologie adaptée à l'environnement de l'Industrie 4.0 et pouvant modéliser rapidement différents aspects d'une entreprise à l'aide de modèles et de construire des systèmes d'aide à la décision pour divers cas d'utilisation. MEMO I4.0 s'adresse aux entreprises ayant une vision à long terme et opérant dans un cadre structuré et normalisé basé sur leurs lignes directives et leur ADN. Les principes clés de MEMO I4.0 sont l'agilité, la modularité, l'interopérabilité et la robustesse. Également, nous fournissons des recommandations pour développer des modèles et un cadre pour construire des systèmes d'aide à la décision conformément à ces principes.

De plus, MEMO I4.0 propose d'utiliser une approche structurée et modulaire. Il s'agit d'un processus en deux étapes organisé selon les phases du cycle de vie pour assurer la robustesse. La première étape, la phase stratégique/tactique, implique le cycle de développement pour créer des modules qui sont utilisés au niveau opérationnel pour construire des modèles intégrés basés sur l'ADN et les lignes directives de l'entreprise et les principes clés. Ces modules nouvellement conÃ§us sont ajoutés à une bibliothèque de modules pour permettre la capitalisation des connaissances. La deuxième étape, la phase opérationnelle, implique le cycle d'exécution pour générer des modèles intégrés et évaluer les performances du système. Un nouveau cycle de développement est déclenché si des modules manquants ou obsolètes sont identifiés. Les modules nécessaires sont ensuite instanciés avec des paramètres spécifiques au cas.

D'autre part, le cadre MEMO I4.0 combine les avantages des cadres de modélisation d'entreprise classiques et des nouveaux cadres proposés pour l'Industrie 4.0. Il comprend quatre dimensions : le cycle de vie, la vue, la généricité et l'échelle. La dimension cycle de vie structure le processus de modélisation en phases pour chaque étape de l'approche structurée, tandis que la dimension vue intègre différents aspects dans les modèles. De plus, les dimensions généricité et échelle sont incluses pour augmenter l'agilité et la flexibilité, qui sont des considérations importantes dans le contexte des usines intelligentes. Plus précisément, la généricité se réfère au degré de réutilisation des modèles développés à l'aide de MEMO I4.0 à différentes entreprises, et l'échelle se réfère au niveau de détail des modèles, donc leur granularité et leur complexité.

Enfin, la méthodologie MEMO I4.0 a été appliquée à trois cas d'utilisation industriels en utilisant des approches de simulation ou de couplage optimisation et simulation pour résoudre différents problèmes de prise de décision. Premièrement, pour évaluer les performances de la logistique interne en évaluant l'impact du changement des politiques d'alimentation de la ligne de production sur les niveaux de stock des composants. Deuxièmement, pour évaluer l'investissement en capacité dans le cadre d'une demande incertaine, et enfin, pour traiter la conception d'une nouvelle ligne d'assemblage pouvant répondre à la demande des clients tout en tenant compte de l'ergonomie. L'application de MEMO I4.0 a permis de répondre rapidement aux besoins des managers et a fourni des informations pertinentes pour la prise de décision dans un environnement incertain où l'agilité est essentielle. Le mécanisme de capitalisation de MEMO I4.0 s'est avéré efficace dans ce contexte.

Dans l'ensemble, ce travail contribue au domaine de la modélisation d'entreprise en fournissant une taxonomie et une cartographie des problèmes de décision dans les systèmes de production, en identifiant les changements dans la prise de décision dans le cadre de l'Industrie 4.0, et en proposant une nouvelle méthodologie de modélisation d'entreprise pouvant soutenir la modélisation et la mise en œuvre des systèmes d'aide à la décision orientés Industrie 4.0. Les futures recherches pourraient généraliser la méthodologie proposée en l'appliquant à divers cas d'utilisation et industries.

Perspectives

Dans ce travail, nous nous sommes concentrés sur l'étude des problèmes de décision dans les systèmes de production et nous avons considéré trois principales activités au sein de l'usine : la production, l'approvisionnement et le stockage. L'une des extensions des résultats sera de suivre la même procédure de trois étapes pour généraliser l'approche à la chaîne d'approvisionnement. Ainsi, d'autres activités avec de nouvelles branches et familles de problèmes seront intégrées avec de nouvelles interactions entre les problèmes de décision.

De plus, il serait intéressant d'explorer la branche de la conception de produits en plus des cinq branches examinées dans notre travail. L'étude des interactions entre les problèmes de décision liés à la conception de produits pourrait s'appuyer sur les recommandations de l'ingénierie simultanée et de l'ingénierie collaborative. Ces approches aident à comprendre comment les décisions de production devraient s'aligner avec les décisions d'ingénierie. De même, étant donné que de nouvelles fonctionnalités de produits sont susceptibles d'être incorporées, il devient crucial de les cartographier efficacement avec les données de gestion du cycle de vie des produits (PLM). Par exemple, la conversion des données de la nomenclature de conception technique (EBOM) issues du PLM en données de nomenclature de production (MBOM) dans le système ERP a déjà été explorée dans la littérature, notamment dans des travaux axés sur l'intégration des modules PLM (Gestion des données du produit) avec ERP.

Actuellement, il est impossible de créer une liste exhaustive des caractéristiques des systèmes de production dans l'environnement de l'Industrie 4.0. Cependant, à mesure que de nouvelles normes pour les concepts de l'Industrie 4.0 seront établies, il sera intéressant de les mettre à jour pour s'aligner avec l'Industrie 4.0 et d'utiliser l'outil de cartographie pour illustrer de nouvelles interactions décisionnelles. Même avec les résultats de la cartographie actuelle, la communauté de recherche peut identifier de nouvelles perspectives et la nécessité de développer de nouveaux modèles intégrés impliquant plusieurs problèmes de décision.

Une voie potentielle pour de futures recherches dans la modélisation d'entreprise dans le cadre de l'Industrie 4.0 consiste à créer des modèles sémantiques ou des ontologies, qui peuvent améliorer les méta-modèles proposés pour la modélisation d'entreprise et représenter efficacement les connexions complexes entre les systèmes cyber-physiques et les jumeaux numériques. Ensuite, nous pourrions examiner comment les systèmes cyber-physiques et les jumeaux numériques peuvent être intégrés pour améliorer la prise de décision et le contrôle en utilisant ces modèles sémantiques ou ontologies pour combler le fossé entre les mondes physique et virtuel. Dans le même sens, davantage de recherche est nécessaire pour utiliser la simulation et l'analyse de processus dans le développement

des systèmes cyber-physiques et des jumeaux numériques, et comment ils peuvent être intégrés dans la modélisation d'entreprise pour soutenir la mise en œuvre des technologies de l'Industrie 4.0.

D'une part, dans le manuscrit, nous présentons la dimension vue et proposons un méta-modèle pour chaque vue afin de soutenir le principe clé d'interopérabilité de MEMO I4.0. L'intégration de ces méta-modèles garantit une cohérence de modélisation. D'autre part, pour l'implémentation, nous proposons d'utiliser la passerelle comme élément physique pour assurer la correspondance avec les différents nœuds de l'entreprise. Une tendance pour améliorer l'interopérabilité dans l'Industrie 4.0 qui doit être explorée est le concept d'un Espace de Noms Unifié (UNS). Il s'agit d'un bus de courtage commun pour la communication d'entreprise utilisant le protocole MQTT. De plus, les différents nœuds de l'entreprise, c'est-à-dire tous les business systèmes et applications métier et les ressources (intelligentes ou non), pourraient être modélisés comme des agents avec un client MQTT pour communiquer via l'UNS et publier leurs valeurs de balise dans des adresses dédiés.

Parmi les contributions de MEMO I4.0 figure la promotion du concept de plug-and-play grâce au principe de modularité dans le développement des système de production dans l'Industrie 4.0. Nous avons suggéré de diviser le système en composants modulaires (modules) composés de trois sous-modules : information, contrôle et physique. Cette approche permet l'interchangeabilité de chaque sous-module au sein d'un module. Une autre perspective à explorer pour atteindre la modularité consiste à utiliser une coquille d'administration pour contrôler, surveiller et gérer les systèmes cyber-physiques.

En ce qui concerne les travaux futurs, il serait intéressant d'incorporer une vue business dans les vues MEMO I4.0 lors de la phase de définition de l'étape stratégique pour formaliser davantage la vision, les lignes directives et l'ADN de l'entreprise, puis valider leur impact sur les processus métiers.

Nous avons déjà constaté que l'achat d'équipements standard proposés dans les catalogues des fournisseurs est devenu peu fréquent. Au lieu de cela, les entreprises industrielles demandent des équipements personnalisés, avec des capacités spécifiques. Dans ce contexte, nous devons généraliser le concept d'aptitude de ressource. Ainsi, il est nécessaire d'introduire un concept d'aptitude générique qui sera incorporé dans le méta-modèle de la vue ressource & organisation.

MEMO I4.0 pourrait servir de référence complète pour l'architecture d'entreprise pour créer une entreprise intelligente dans l'environnement de l'Industrie 4.0. Cependant, des recherches concernant les outils, les langages et les ontologies conformes au cadre MEMO I4.0 sont nécessaires. De plus, pour évaluer l'impact de l'Industrie 4.0 dans un contexte différent, nous prévoyons que la communauté de recherche et les praticiens appliqueront MEMO I4.0 dans d'autres domaines, tels que la logistique, l'agriculture et les soins de santé.

Nous avons constaté que trois techniques sont les plus utilisées pour développer des systèmes de prise de décision : l'optimisation, la simulation et les techniques d'apprentissage automatique. Dans notre manuscrit, les cas d'utilisation pour lesquels MEMO I4.0 a été appliqué se sont concentrés sur l'optimisation et la simulation, ainsi que leur intégration. Les travaux futurs pourraient étendre l'utilisation de MEMO I4.0 et étudier l'utilisation des techniques d'apprentissage automatique pour améliorer les processus de prise de décision ou la combinaison de l'apprentissage automatique avec l'optimisation et la simulation pour dériver des systèmes d'aide à la décision basés sur l'apprentissage automatique, la simulation et l'optimisation. Pour cela, il sera nécessaire de mettre à jour la feuille de route proposée pour développer les modules.

Enfin, nous avons identifié la nécessité d'étudier les implications de l'intégration des systèmes IT (Technologie de l'Information) et OT (Technologie Opérationnelle) et comment concevoir une architecture pour la prendre en charge de manière efficace et efficiente, c'est-à-dire qui puisse gérer la complexité et la dynamique des systèmes de l'Industrie 4.0 et améliorer la scalabilité, la sécurité et la fiabilité. L'idée principale est de trouver l'infrastructure et les protocoles de réseau pour gérer le volume élevé de données générées par les systèmes cyber-physiques. Par exemple, Kafka est utilisé dans le cloud, rarement en Edge en tant que système de messagerie pour gérer les événements et soutenir la prise de décision basée sur les données. Par conséquent, il est nécessaire de mener davantage des recherches sur les exigences en matière d'architecture réseau pour l'Industrie 4.0 et comment elles peuvent être intégrées dans la modélisation d'entreprise pour soutenir la mise en œuvre des technologies de l'Industrie 4.0. De plus, de nouvelles recherches se concentrant sur les technologies émergentes telles que le calcul Edge, le calcul Fog et le calcul distribué dans l'industrie 4.0 sont appréciées pour soutenir l'architecture IT/OT.

Dans l'ensemble, ce travail ouvre des perspectives pour la recherche future dans le domaine de l'Industrie 4.0 et de la modélisation d'entreprise. Les différentes extensions possibles de la méthodologie MEMO I4.0, l'exploration de nouvelles applications dans différents secteurs et l'intégration de nouvelles technologies prometteuses sont autant de domaines qui pourraient bénéficier de recherches futures. L'Industrie 4.0 étant un domaine en constante évolution, la communauté de recherche doit continuer à collaborer et à innover pour relever les défis et exploiter les opportunités offertes par cette révolution industrielle. De plus, avec le nouveaux travaux impliquant le concept d'Industrie 5.0, la capacité de MEMO I4.0 à faire face aux défis imposés par ce concept pourrait être testée, et une extension pourrait être possible si nécessaire.

Conclusion & Perspectives

In conclusion, this manuscript starts by identifying the challenges and drivers of the transformation from traditional Manufacturing Systems (MSs) to Industry 4.0 and the resulting changes in the decision-making process. Then, it summarizes this transformation into three evolutions: technical, organizational, and transversal.

The technical evolution concerns the drivers of change in MSs in the context of Industry 4.0, including standardization of data exchange, and new techniques and algorithms. Standardization of data exchange is necessary to ensure consistency and interoperability in data exchange between components. On the other hand, new techniques and algorithms, such as big data and machine learning, are needed to deal with the large amounts of data generated by new technologies and improve decision-making.

The organizational evolution concerns the emergence of new revenue models, such as dynamic pricing and pay-by-usage, and the need for companies to adapt and optimize their MSs to be agile and lean to meet customer requirements. The concept of "leagility", which combines leanness and agility, is introduced as a way for companies to adopt this organizational mode. In addition, there is a need for developing new skills and employment activities due to the emergence of new technologies in industry 4.0 and the importance of upskilling or reskilling employees to adapt to these changes. In response to the uncertainty imposed by the VUCA world, companies must move towards increasingly adaptable production systems. The leagile strategy, the human-centered MS, as well as the adaptable MS are presented as three organizational evolutions that address the evolution of business models, the development of new skills, and the uncertainty in decisions made in the context of Industry 4.0, respectively. The leagile strategy involves implementing postponement strategies and using lean principles for standard products and agile principles for customized products, while the human-centered MS focuses on putting the human factor at the center of the MS and improving the working environment and training programs for employees. Finally, the adaptable production systems make it possible to bring a new feature of self-organization and autonomy of decision, allowing them to face the hazards.

The transversal evolution concerns the development of new Key Performance Indicators (KPIs), the real-time monitoring, and the decentralized control. First, the concept of KPIs is highlighted and their role in understanding and improving the performance of

MSs. Then, the concept of a Performance Measurement System (PMS) is introduced, it is a set of metrics used to quantify the efficiency and effectiveness of MSs. Classical KPIs in MSs are divided into cost, time, and quality. However, in the context of industry 4.0, new KPIs, known as KPIs 4.0, are needed to evaluate the performance of MSs due to changes in modeling and behavior, the evolution of time horizons of decision problems, and their interactions. These new KPIs are divided into sustainability and agility. Therefore, it is essential that the development of a PMS within Industry 4.0 integrates multiple perspectives when defining KPIs and can handle the complexity and dynamics of industry 4.0. Real-time monitoring involves using smart objects that can monitor themselves and request maintenance or assistance when needed, improving efficiency, extending component lifespans, and enabling decision-making. The decentralized control approach involves distributing operations across different nodes to cope with the complexity of a smart factory and enables self-organization and flexibility.

Afterward, the manuscript proposed a taxonomy of decision problems in MSs and their interactions, as well as the evolution of decision-making in the context of Industry 4.0. Through the proposed three-step procedure, we were able to structure and map decision problems in MSs. The first step involves identifying decision problems in MSs, which are classified into three categories: production, supply, and storage. The second step involves the definition of decision problems based on the CIAOC (Constraints, Inputs, Assumptions, Outputs, Criteria) properties. Finally, the third step involves the study of the interactions among the decision problems, which is carried out through causality and feedback loops. A cartography of decision problems is proposed, which represents the relationships and dependencies among the identified decision problems. It provides a starting point for researchers and practitioners to consider the decision problems, their interactions in MSs, and the changes in decision-making within Industry 4.0 and afterward.

Based on our deep analysis of the transformation of MSs within Industry 4.0, mainly the decision-making, we proposed a new Methodology of Enterprise Modeling Oriented I4.0, called MEMO I4.0, to address the lack of an approach that can support the modeling and implementation of Industry 4.0-oriented decision support systems. The proposed methodology was applied to derive a modern decision support system adapted to Industry 4.0 through case studies, which showed its efficiency and effectiveness in this context.

MEMO I4.0 is a data-driven approach that aims to create a digital twin of an MS that is scalable, reconfigurable, and uses near real-time data. The ultimate goal of MEMO I4.0 is to offer a methodology adapted to the Industry 4.0 environment and can quickly model different aspects of an enterprise using models and build decision support systems for various use cases. MEMO I4.0 is intended for companies with a long-term vision and operating in a structured, standardized framework based on their guidelines and DNA. The key principles of MEMO I4.0 are agility, modularity, interoperability, and robustness.

Likewise, we provide recommendations for developing models and a framework for building decision support systems according to these principles.

Furthermore, MEMO I4.0 proposes using a structured, modular approach. It is a two-stage process organized according to the life cycle phases to ensure robustness. The first stage, the strategical/tactical stage, involves the development cycle to create modules that are used at the operational level to build integrated models based on the enterprise's DNA/guidelines and key principles. These newly conceived modules are added to a library of modules to allow for knowledge capitalization. The second stage, the operational stage, involves the execution cycle to generate integrated models and assess the system's performance. A new development cycle is triggered if missing or obsolete modules are identified. The required modules are then instantiated with case-specific parameters.

On the other hand, the MEMO I4.0 framework combines the advantages of classical enterprise modeling frameworks and new frameworks proposed for Industry 4.0. It includes four dimensions: life cycle, view, genericity, and scale. The life cycle dimension structures the modeling process into phases for each stage of the structured approach, while the view dimension integrates different aspects into the models. In addition, the dimensions of genericity and scale are included to increase agility and flexibility, which are important considerations in the context of smart factories. To be precise, the genericity refers to the degree to which the models developed using MEMO I4.0 can be applied to different enterprises, and the scale refers to the level of detail of the models, thus, their granularity and complexity.

Finally, MEMO I4.0 methodology was applied in three industrial use cases using simulation or simulation-optimization approaches to address various decision-making problems. First, to assess internal logistics performance by evaluating the impact of changing line feeding policies on component inventory levels. Second, to evaluate the capacity investment in uncertain demand, and lastly, to deal with the design of a new assembly line that can meet customer demand while considering ergonomics. Applying MEMO I4.0 allowed for quick responses to managers' needs and provided valuable insights for decision-making in an uncertain environment where agility is essential. The capitalization mechanism of MEMO I4.0 proved to be effective in this context.

Overall, this work contributes to the field of enterprise modeling by providing a taxonomy and cartography of decision problems in MSs, identifying the changes in decision-making within Industry 4.0, and proposing a new enterprise modeling methodology that can support the modeling and implementation of Industry 4.0-oriented decision support systems.

Future research could generalize the proposed methodology by applying it to various case studies and industries.

Perspectives

In this work, we focused on investigating the decision problems in manufacturing systems (MSs), and we considered three main activities within the plant: production, supply, and storage. One of the extensions of the findings will be to follow the same three-step procedure to generalize the approach for the supply chain. Therefore, other activities with new branches and problem families will be integrated with new interactions between decision problems.

Potential area for future research is to explore the product design branch in addition to the five branches examined in our work. Investigating the interactions of decision problems related to product design can draw upon the recommendations of concurrent engineering and collaborative engineering. These approach help in understanding how manufacturing decisions should align with engineering ones. Furthermore, as additional product features are probably to be incorporated, it becomes crucial to map them effectively with Product Lifecycle Management (PLM) data. For instance, converting Engineering Bill Of Materials (EBOM) data from PLM to Manufacturing Bill Of Materials (MBOM) data in ERP has been previously explored in the literature, especially in works focused on integrating PLM modules (Product Data and Management (PDM)) with ERP.

Currently, creating a comprehensive list of the MS features of MSs in the Industry 4.0 environment is impossible. However, as new standardizations for Industry 4.0 concepts are established, updating them to align with Industry 4.0 and using the cartography tool to illustrate new decision interactions will be worthwhile. Note that even with the result of the current cartography, the research community can capture new insights and the need for new integrated model development involving several decision problems.

A potential avenue for further research in enterprise modeling within Industry 4.0 is to create semantic models or ontologies, which can enhance the proposed metamodels for enterprise modeling and effectively depict the intricate connections between cyber-physical systems and digital twins. After that, we could examine how cyber-physical systems and digital twins can be integrated to improve decision-making and control by using these semantic models or ontologies to bridge the gap between the physical and virtual worlds. In the same direction, more research is needed to use simulation and process mining in developing cyber-physical systems and digital twins and how they can be integrated into enterprise modeling to support the implementation of Industry 4.0 technologies.

On the one hand, we present in the manuscript the view dimension and propose a metamodel of each view to support the interoperability key principle of MEMO I4.0. The integration of these metamodels ensures modeling consistency. On the other hand, for the implementation, we propose using the gateway as a physical element to ensure the mapping with different enterprise nodes. One trend to improve interoperability within

the Industry 4.0 that must be investigated is the concept of a unified namespace (UNS). It consists of a common broker bus for enterprise communication using the MQTT protocol. Furthermore, the different nodes of the enterprise, i.e., all business systems and the resources (smart or not), could be modeled as agents with an MQTT client to communicate through the UNS and publish their tags in dedicated topics.

Among MEMO I4.0 contributions are the promotion of the plug-and-play concept through the modularity principle in developing MS within Industry 4.0. We suggested breaking down the system into modular components (modules) consisting of three sub-modules: information, control, and physical. This approach allows for the interchangeability of each sub-module within a module. Another perspective to explore to achieve modularity is utilizing an administration shell to control, monitor, and manage cyber-physical systems.

As to future work, it would be interesting to incorporate a business view into the MEMO I4.0 views during the definition phase of the strategical stage to formalize better the company's vision, guidelines, and DNA, then validate their impact on business processes.

We have already noted that purchasing standard equipment proposed in the vendor catalogs has become infrequent. Instead, industrial companies ask for personalized equipment, including specific capabilities. In this context, we need to generalize the concept of resource capability. Thus there is necessary to introduce a concept of generic capability that will be incorporated into the metamodel of the resource & organization view.

MEMO I4.0 could serve as a comprehensive reference enterprise architecture for creating a smart enterprise within the Industry 4.0 environment. However, research concerning tools, languages, and ontologies compliant with the MEMO I4.0 framework is required. In addition, to evaluate the impact of Industry 4.0 in a different context, we expect that the research community and the practitioners will apply MEMO I4.0 in other domains, such as logistics, agriculture, and healthcare.

We found that three techniques are the most used to develop decision support systems: optimization, simulation, and machine learning (ML) techniques. In our manuscript, the use cases for which MEMO I4.0 were applied, focused on optimization and simulation and their integration. Future works could extend the use of MEMO I4.0 and investigate the use of ML techniques to improve decision-making processes or the combination of ML with optimization and simulation to derive the ML-Simulation-Optimization-based DSSs. To do so, updating the roadmap proposed to develop modules will be necessary.

Finally, we identified the need to investigate the implications of the integration of IT (Information Technology) and OT(Operational Technology) systems and how to design the architecture to support it effectively and efficiently, i.e., that can support the complexity and dynamic of Industry 4.0 systems and improve the scalability, security and reliability. The main idea is to find the network infrastructure and protocols to

handle the high volume of data generated by cyber-physical systems. For instance, Kafka is used in the cloud and at the edge as a messaging system to handle events and support data-driven decision-making. Therefore, we need more effort on the network architecture requirements for Industry 4.0 and how they can be integrated into enterprise modeling to support the implementation of Industry 4.0 technologies. Furthermore, new research focusing on emerging technologies like Edge computing, Fog computing, and distributed computing in industry 4.0 are appreciated to support the IT/OT architecture.

Overall, this work opens perspectives for future research in Industry 4.0 and enterprise modeling. The various possible extensions of the MEMO I4.0 methodology, the exploration of new applications in different sectors, and the integration of promising new technologies are all areas that could benefit from future research. As Industry 4.0 is an ever-changing field, the research community must continue to collaborate and innovate to meet the challenges and exploit the opportunities offered by this industrial revolution. Moreover, with the new works involving the concept of Industry 5.0, the ability of MEMO I4.0 to face challenges imposed by this concept could be tested, and an extension might be possible if needed.

References

Abdallah, A. B. and M. Yoshiki (2009). The impact of lean practices on mass customization and competitive performance of mass-customizing plants. In *Proceedings of the 20th Annual Production and Operations Management Society (POMS) Conference*, Volume 5.

Abdi, M. R. and A. W. Labib (2004). Grouping and selecting products: the design key of reconfigurable manufacturing systems (RMSs). *International Journal of Production Research 42*(3), 521–546.

Abdous, M. A. (2019). *Optimal design of manufacturing systems with ergonomics: application to assembly lines*. Ph. D. thesis, Mines Saint-Etienne (France) and The University of Padua (Italy).

Abdous, M. A., X. Delorme, D. Battini, F. Sgarbossa, and S. Berger-Douce (2018a). Assembly line balancing problem with consideration of workers fatigue and recovery. *20th International Working Seminar on Production Economics - IWSPE 51*(11), 12.

Abdous, M. A., X. Delorme, D. Battini, F. Sgarbossa, and S. Berger-Douce (2018b). Multi-objective optimization of assembly lines with workers fatigue consideration. *IFAC-PapersOnLine 51*(11), 698 – 703.

Ahmad, M. M. and N. Dhafr (2002). Establishing and improving manufacturing performance measures. *Robotics and Computer-Integrated Manufacturing 18*(3-4), 171–176.

Akcali, E. and R. Uzsoy (2000). A sequential solution methodology for capacity allocation and lot scheduling problems for photolithography. In *Twenty Sixth IEEE/CPMT International Electronics Manufacturing Technology Symposium (Cat. No.00CH37146)*. IEEE.

Al-Shayea, A. M. (2012). Maintenance capacity planning: Determination of maintenance workforce. *Engineering 04*(01), 37–43.

Albo, A. and P. Falkman (2020). A standardization approach to virtual commissioning strategies in complex production environments. *Procedia Manufacturing 51*, 1251–1258.

Alfathi, N., A. Lyhyaoui, and A. Sedqui (2019, nov). Fusion of dedicated and shared storage to maximize the use of space of static warehouses. *Journal of Industrial and Production Engineering 36*(8), 512–532.

Alnahhal, M. and B. Noche (2015a). Dynamic material flow control in mixed model assembly lines. *Computers & Industrial Engineering 85*, 110–119.

Alnahhal, M. and B. Noche (2015b). A genetic algorithm for supermarket location problem. *Assembly Automation 35*(1), 122–127.

Anderson, T. W. (2011). Anderson–darling tests of goodness-of-fit. In *International Encyclopedia of Statistical Science*, pp. 52–54. Springer Berlin Heidelberg.

Ang, M., Y. F. Lim, and M. Sim (2012). Robust storage assignment in unit-load warehouses. *Management Science 58*(11), 2114–2130.

Anglani, A., A. Grieco, M. Pacella, and T. Tolio (2002). Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure. *Simulation Modelling Practice and Theory 10*(3-4), 209–234.

Ante, G., F. Facchini, G. Mossa, and S. Digiesi (2018). Developing a key performance indicators tree for lean and smart production systems. *IFAC-PapersOnLine 51*(11), 13–18.

Arnott, D. and G. Dodson (2008). Decision support systems failure. In *Handbook on Decision Support Systems 1*, pp. 763–790. Springer Berlin Heidelberg.

Arnott, D. and G. Pervan (2014). A critical analysis of decision support systems research revisited: The rise of design science. *Journal of Information Technology 29*(4), 269–293.

Atan, Z., T. Ahmadi, C. Stegehuis, T. De Kok, and I. Adan (2017). Assemble-to-order systems: A review. *European Journal of Operational Research 261*(3), 866–879.

Atzori, L., A. Iera, and G. Morabito (2010). The internet of things: A survey. *Computer Networks 54*(15), 2787–2805.

Bader, S., M. Maleshkova, and S. Lohmann (2019). Structuring reference architectures for the industrial internet of things. *Future Internet 11*(7), 151.

Baily, M. N., J. Manyika, and S. Gupta (2013). US productivity growth: An optimistic perspective. *International Productivity Monitor 25*, 3–12.

Baines, T. S., H. W. Lightfoot, S. Evans, A. Neely, R. Greenough, J. Peppard, R. Roy, E. Shehab, A. Braganza, A. Tiwari, J. R. Alcock, J. P. Angus, M. Bastl,

A. Cousens, P. Irving, M. Johnson, J. Kingston, H. Lockett, V. Martinez, P. Michele, D. Tranfield, I. M. Walton, and H. Wilson (2007). State-of-the-art in product-service systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 221*(10), 1543–1552.

Baller, R., S. Hage, P. Fontaine, and S. Spinler (2020). The assembly line feeding problem: An extended formulation with multiple line feeding policies and a case study. *International Journal of Production Economics 222*, 107489.

Barbieri, L., C. Mussida, M. Piva, and M. Vivarelli (2020). Testing the employment and skill impact of new technologies. In *Handbook of Labor, Human Resources and Population Economics*, pp. 1–27. Springer International Publishing.

Bartolini, M., E. Bottani, and E. H. Grosse (2019). Green warehousing: Systematic literature review and bibliometric analysis. *Journal of Cleaner Production 226*, 242–258.

Bassan, Y., Y. Roll, and M. J. Rosenblatt (1980). Internal layout design of a warehouse. *A I I E Transactions 12*(4), 317–322.

Battaïa, O. and A. Dolgui (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics 142*(2), 259–277.

Battini, D., M. Calzavara, A. Otto, and F. Sgarbossa (2016). The integrated assembly line balancing and parts feeding problem with ergonomics considerations. *IFAC-PapersOnLine 49*(12), 191–196.

Battini, D., M. Calzavara, A. Persona, and F. Sgarbossa (2016). Additional effort estimation due to ergonomic conditions in order picking systems. *International Journal of Production Research 55*(10), 2764–2774.

Battini, D., M. Faccio, E. Ferrari, A. Persona, and F. Sgarbossa (2007). Design configuration for a mixed-model assembly system in case of low product demand. *The International Journal of Advanced Manufacturing Technology 34*(1), 188–200.

Battini, D., M. Faccio, A. Persona, and F. Sgarbossa (2009). Design of the optimal feeding policy in an assembly system. *International Journal of Production Economics 121*(1), 233–254.

Battini, D., M. Faccio, A. Persona, and F. Sgarbossa (2010). "supermarket warehouses": stocking policies optimization in an assembly-to-order environment. *The International Journal of Advanced Manufacturing Technology 50*(5-8), 775–788.

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science 32*(8), 909–932.

Becker, C. and A. Scholl (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research 168*(3), 694–715.

Belassiria, I., M. Mazouzi, S. ELfezazi, A. Cherrafi, and Z. ELMaskaoui (2018). An integrated model for assembly line re-balancing problem. *International Journal of Production Research 56*(16), 5324–5344.

Ben-Daya, M., U. Kumar, and D. N. P. Murthy (2016). *Maintenance Planning, Scheduling, and Control*, Chapter 19, pp. 451–473. John Wiley & Sons, Ltd.

Bencak, P., D. Hercog, and T. Lerher (2022). Indoor positioning system based on bluetooth low energy technology and a nature-inspired optimization algorithm. *Electronics 11*(3), 308.

Benjaafar, S. and M. ElHafsi (2006). Production and inventory control of a single product assemble-to-order system with multiple customer classes. *Management Science 52*(12), 1896–1912.

Bennara, M., A. Zimmermann, M. Lefrançois, and N. Messalti (2020). Interoperability of semantically-enabled web services on the WoT. In *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services*. ACM.

Berio, G. and F. Vernadat (2001). Enterprise modelling with CIMOSA: Functional and organizational aspects. *Production Planning & Control 12*(2), 128–136.

Bernus, P., O. Noran, and A. Molina (2014). Enterprise architecture: Twenty years of the GERAM framework. *IFAC Proceedings Volumes 47*(3), 3300–3308.

Bevilacqua, M. and M. Braglia (2000). The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety 70*(1), 71–83.

Bhamu, J. and K. Singh Sangwan (2014). Lean manufacturing: literature review and research issues. *International Journal of Operations & Production Management 34*(7), 876–940.

Bishop, C. M. (1994). Neural networks and their applications. *Review of Scientific Instruments 65*(6), 1803–1832.

Biswas, S. and Y. Narahari (2004). Object oriented modeling and decision support for supply chains. *European Journal of Operational Research 153*(3), 704–726.

Bollapragada, R., U. S. Rao, and J. Zhang (2004). Managing inventory and supply performance in assembly systems with random supply capacity and demand. *Management Science 50*(12), 1729–1743.

Borg, G. (1982). Psychophysical bases of perceived exertion. *Medicine and Science in Sports and Exercise 14*(5), 377–381.

Borges, A. F. S., F. J. B. Laurindo, M. M. Spínola, R. F. Gonçalves, and C. A. Mattos (2021). The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions. *International Journal of Information Management 57*, 102225.

Borshchev, A. (2014). Multi-method modelling: AnyLogic. In *Discrete-Event Simulation and System Dynamics for Management Decision Making*, pp. 248–279. John Wiley & Sons Ltd.

Bortolini, M., L. Botti, F. G. Galizia, and A. Regattieri (2021). Bi-objective design and management of reconfigurable manufacturing systems to optimize technical and ergonomic performances. *Applied Sciences 11*(1), 263.

Bortolini, M., M. Faccio, G. F. G., M. Gamberi, and F. Pilati (2021). Adaptive automation assembly systems in the industry 4.0 era: A reference framework and full–scale prototype. *Applied Sciences 11*(3), 1256.

Bortolini, M., E. Ferrari, F. G. Galizia, C. Mora, and F. Pilati (2019). Optimal redesign of cellular flexible and reconfigurable manufacturing systems. *Procedia CIRP 81*, 1435–1440.

Bortolini, M., E. Ferrari, M. Gamberi, F. Pilati, and M. Faccio (2017). Assembly system design in the industry 4.0 era: a general framework. *IFAC-PapersOnLine 50*(1), 5700–5705.

Bougrine, A., S. Darmoul, and S. Hajri-Gabouj (2017). TOPSIS based multi-criteria reconfiguration of manufacturing systems considering operational and ergonomic indicators. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. IEEE.

Bouh, M. A. and D. Riopel (2016). *Selection of Material Handling and Warehousing Equipment: Data Analysis Process and Statistics*. Interuniversity Research Center on Enterprise Networks, Logistics and Transportation.

Boysen, N., D. Briskorn, and S. Emde (2017). Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research 259*(1), 293–307.

Boysen, N., S. Emde, M. Hoeck, and M. Kauderer (2015). Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research 242*(1), 107–120.

Boysen, N., M. Fliedner, and A. Scholl (2007). A classification of assembly line balancing problems. *European Journal of Operational Research 183*(2), 674–693.

Boysen, N., M. Fliedner, and A. Scholl (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics 111*(2), 509–528.

Boysen, N., M. Fliedner, and A. Scholl (2009). Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research 192*(2), 349–373.

Bozer, Y. A. and J. A. White (1984). Travel-time models for automated storage/retrieval systems. *IIE Transactions 16*(4), 329–338.

Brailsford, S. C., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio (2019). Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research 278*(3), 721–737.

Brettel, M., N. Friederichsen, M. Keller, and M. Rosenberg (2014). How virtualization , decentralization and network building change the manufacturing landscape : An industry 4.0. *International Journal of Information and Communication Engineering 8*(1), 37 – 44.

Briand, C., Y. He, and S. U. Ngueveu (2018). Energy-efficient planning for supplying assembly lines with vehicles. *EURO Journal on Transportation and Logistics 7*(4), 387–414.

Budai, G., R. Dekker, and R. P. Nicolai (2008). Maintenance and production: A review of planning models. In *Springer Series in Reliability Engineering*, pp. 321–344. Springer London.

Bueno, A., M. G. Filho, and A. G. Frank (2020). Smart production planning and control in the industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering 149*, 106774.

Calzavara, M., C. H. Glock, E. H. Grosse, A. Persona, and F. Sgarbossa (2017). Analysis of economic and ergonomic performance measures of different rack layouts in an order picking warehouse. *Computers & Industrial Engineering 111*, 527–536.

Campana, N. P., M. Iori, and M. C. O. Moreira (2021). Mathematical models and heuristic methods for the assembly line balancing problem with hierarchical worker assignment. *International Journal of Production Research 0*(0), 1–19.

Cano, J. A., A. A. Correa-Espinal, and R. A. Gómez-Montoya (2017). An evaluation of picking routing policies to improve warehouse efficiency. *International Journal of Industrial Engineering and Management 8*(4), 229–238.

Cao, W., P. Jiang, B. Liu, and K. Jiang (2017). Real-time order scheduling and execution monitoring in public warehouses based on radio frequency identification. *The International Journal of Advanced Manufacturing Technology 95*(5-8), 2473–2494.

Cardona, L. F., D. F. Soto, L. Rivera, and H. J. Martínez (2015). Detailed design of fishbone warehouse layouts with vertical travel. *International Journal of Production Economics 170*, 825–837.

Çelk, M. and H. Süral (2013). Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Transactions 46*(3), 283–300.

Cergibozan, Ç. and A. S. Tasan (2016). Order batching operations: an overview of classification, solution techniques, and future research. *Journal of Intelligent Manufacturing 30*(1), 335–349.

Cerqueus, A. and X. Delorme (2023). Evaluating the scalability of reconfigurable manufacturing systems at the design phase. *International Journal of Production Research 0*(0), 1–14.

Ceryan, O. and Y. Koren (2009). Manufacturing capacity planning strategies. *CIRP annals 58*(1), 403–406.

CFAI Loire and Institut Henri Fayol de Mines Saint-Etienne (2018). It'm factory platform.

Chakrabortty, R. K. and M. A. A. Hasin (2013). Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach. *International Journal of Industrial Engineering Computations 4*(1), 1–12.

Chakraborty, S. and D. Banik (2005). Design of a material handling equipment selection model using analytic hierarchy process. *The International Journal of Advanced Manufacturing Technology 28*(11-12), 1237–1245.

Chang, D.-T. and U. P. Wen (1997). The impact on rack configuration on the speed profile of the storage and retrieval machine. *IIE Transactions 29*(7), 525–531.

Chang, K.-H., Y.-H. Huang, and S.-P. Yang (2013). Vehicle fleet sizing for automated material handling systems to minimize cost subject to time constraints. *IIE Transactions 46*(3), 301–312.

Chaudhuri, A., P. P. Datta, K. J. Fernandes, and Y. Xiong (2021). Optimal pricing strategies for manufacturing-as-a service platforms to ensure business sustainability. *International Journal of Production Economics 234*, 108065.

Chen, D. and G. Doumeingts (1996). The GRAI-GIM reference model, architecture and methodology. In P. Bernus, L. Nemes, and T. J. Williams (Eds.), *Architectures for Enterprise Integration*, pp. 102–126. Boston, MA: Springer US.

Chen, T.-L., C.-Y. Cheng, Y.-Y. Chen, and L.-K. Chan (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics 159*, 158–167.

Chengalur, S. N. (2004). *Kodak's ergonomic design for people at work*. John Wiley & sons.

Chibani, A., X. Delorme, A. Dolgui, and H. Pierreval (2018). Dynamic optimisation for highly agile supply chains in e-procurement context. *International Journal of Production Research 56*(17), 5904–5929.

Choi, W. and Y. Lee (2002). A dynamic part-feeding system for an automotive assembly line. *Computers & Industrial Engineering 43*(1-2), 123–134.

Chryssolouris, G. (1996). Flexibility and its measurement. *CIRP Annals 45*(2), 581–587.

Chryssolouris, G. (2006). *Manufacturing Systems: Theory and Practice, 2nd Edition*, Chapter 5, pp. 329–342. Springer-Verlag.

Cohen, Y., H. Naseraldin, A. Chaudhuri, and F. Pilati (2019). Assembly systems in industry 4.0 era: a road map to understand assembly 4.0. *The International Journal of Advanced Manufacturing Technology 105*(9), 4037–4054.

Cormier, G. and E. A. Gunn (1996). On coordinating warehouse sizing, leasing and inventory policy. *IIE Transactions 28*(2), 149–154.

Costabile, G., M. Fera, F. Fruggiero, A. Lambiase, and D. Pham (2017). Cost models of additive manufacturing: A literature review. *International Journal of Industrial Engineering Computations 8*(2), 263–283.

Csalódi, R., Z. Süle, S. Jaskó, T. Holczinger, and J. Abonyi (2021). Industry 4.0-driven development of optimization algorithms: A systematic overview. *Complexity 2021*, 1–22.

Custodio, L. and R. Machado (2019, nov). Flexible automated warehouse: a literature review and an innovative framework. *The International Journal of Advanced Manufacturing Technology volume 106*(1-2), 533–558.

De Koster, R. B. M., T. Le-Duc, and K. J. Roodbergen (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research 182*(2), 481–501.

De Koster, R. B. M., T. Le-Duc, and N. Zaerpour (2012). Determining the number of zones in a pick-and-sort order picking system. *International Journal of Production Research 50*(3), 757–771.

Deif, A. M. and W. ElMaraghy (2006). Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system. *The International Journal of Advanced Manufacturing Technology 32*(5-6), 557–562.

Delorme, X., O. Battaïa, and A. Dolgui (2013). Multi-objective approaches for design of assembly lines. In *Springer Series in Advanced Manufacturing*, pp. 31–56. Springer London.

Demir, K. A., G. DÃ¶ven, and B. Sezen (2019). Industry 5.0 and human-robot co-working. *Procedia Computer Science 158*, 688–695.

Demir, Y. and S. K. İşleyen (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling 37*(3), 977–988.

Derhami, S., J. S. Smith, and K. R. Gue (2016). Optimising space utilisation in block stacking warehouses. *International Journal of Production Research 55*(21), 6436–6452.

Diefenbach, H., S. Emde, and C. H. Glock (2020). Loading tow trains ergonomically for just-in-time part supply. *European Journal of Operational Research 284*(1), 325–344.

Dillon, S. M. (1998). Descriptive decision making: Comparing theory with practice. In *Proceedings of 33rd ORSNZ Conference, University of Auckland, New Zealand.*

Djuric, A. M., R. J. Urbanic, and J. L. Rickli (2016). A framework for collaborative robot (CoBot) integration in advanced manufacturing systems. *SAE International Journal of Materials and Manufacturing 9*(2), 457–464.

Dolgui, A., D. Ivanov, S. P. Sethi, and B. Sokolov (2018). Scheduling in production, supply chain and industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International Journal of Production Research 57*(2), 411–432.

Doumeingts, G., D. Chen, B. Vallespir, P. Fenie, and F. Marcotte (1993). GIM (GRAI Integrated Methodology) and its evolutions-a methodology to design and specify advanced manufacturing systems. In *DIISM*, Volume 93, pp. 101–120.

Doumeingts, G. and Y. Ducq (2001). Enterprise modelling techniques to improve efficiency of enterprises. *Production Planning & Control 12*(2), 146–163.

Drahos, P., E. Kucera, O. Haffner, and I. Klimo (2018). Trends in industrial communication and OPC UA. In *2018 Cybernetics & Informatics (K&I)*. IEEE.

Drath, R. and A. Horch (2014). Industrie 4.0: Hit or hype? [industry forum]. *IEEE Industrial Electronics Magazine 8*(2), 56–58.

Drira, A., H. Pierreval, and S. Hajri-Gabouj (2007). Facility layout problems: A survey. *Annual Reviews in Control 31*(2), 255–267.

Duffuaa, S. O. and A. Raouf (2015). *Planning and Control of Maintenance Systems*. Springer International Publishing.

Edmondson, N. F. and A. H. Redford (2002). Generic flexible assembly system design. *Assembly Automation 22*(2), 139–152.

Egger, J. and T. Masood (2020). Augmented reality in support of intelligent manufacturing – a systematic literature review. *Computers & Industrial Engineering 140*, 106195.

Ejsmont, K. (2021). The impact of industry 4.0 on employees—insights from australia. *Sustainability 13*(6), 3095.

El Alaoui El Abdellaoui, M., E. Bricard, G. f., P. Gianessi, and X. Delorme (2020). Scalable, reconfigurable simulation models in industry 4.0-oriented enterprise modeling. In *2020 Winter Simulation Conference (WSC)*. IEEE.

El Alaoui El Abdellaoui, M., F. Grimaud, P. Gianessi, and X. Delorme (2019). Integrated decision process to design manufacturing systems towards industry 4.0. *IFAC-PapersOnLine 52*(13), 1373–1378.

El Zaatari, S., M. Marei, W. Li, and Z. Usman (2019). Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems 116*, 162–180.

Elbert, R. M., T. Franzke, C. H. Glock, and E. H. Grosse (2017). The effects of human behavior on the efficiency of routing policies in order picking: The case of route deviations. *Computers & Industrial Engineering 111*, 537–551.

Elkington, J. (1997). *Cannibals With Forks: The Triple Bottom Line of 21st Century Business*, Chapter 1, pp. 15–40. Capstone.

ElMaraghy, H. A. (2005). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems 17*(4), 261–276.

Emde, S. and N. Boysen (2011). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *European Journal of Operational Research 217*(2), 287–299.

Emde, S. and N. Boysen (2012a). Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *International Journal of Production Economics 135*(1), 393–402.

Emde, S. and N. Boysen (2012b). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *European Journal of Operational Research 217*(2), 287–299.

Emde, S., N. Boysen, and A. Scholl (2009). Balancing mixed-model assembly lines: a computational evaluation of objectives to smoothen workload. *International Journal of Production Research 48*(11), 3173–3191.

Emde, S., M. Fliedner, and N. Boysen (2012). Optimally loading tow trains for just-in-time supply of mixed-model assembly lines. *IIE Transactions 44*(2), 121–135.

Emde, S. and M. Gendreau (2017). Scheduling in-house transport vehicles to feed parts to automotive assembly lines. *European Journal of Operational Research 260*(1), 255–267.

Emde, S. and M. Schneider (2018). Just-in-time vehicle routing for in-house part feeding to assembly lines. *Transportation Science 52*(3), 657–672.

Ene, S., İ. KÃ¼çÃ¼koğlu, A. Aksoy, and N. ÖztÃ¼rk (2016). A genetic algorithm for minimizing energy consumption in warehouses. *Energy 114*, 973–980.

Ene, S. and N. Öztürk (2011). Storage location assignment and order picking optimization in the automotive industry. *The International Journal of Advanced Manufacturing Technology 60*(5-8), 787–797.

Eom, S. B. (1999). Decision support systems research: current state and trends. *Industrial Management & Data Systems 99*(5), 213–221.

Er, M. C. (1988). Decision support systems: A summary, problems, and future trends. *Decision Support Systems 4*(3), 355–363.

Erol, S., A. JÃ¤ger, P. Hold, K. Ott, and W. Sihn (2016). Tangible industry 4.0: A scenario-based approach to learning for the future of production. *Procedia CIRP 54*, 13–18.

Eslami, Y., M. Dassisti, M. Lezoche, and H. Panetto (2018). A survey on sustainability in manufacturing organisations: dimensions and future insights. *International Journal of Production Research 57*(15-16), 5194–5214.

Etz, D., H. Brantner, and W. Kastner (2020). Smart manufacturing retrofit for brownfield systems. *Procedia Manufacturing 42*, 327–332.

Faccio, M., M. Gamberi, A. Persona, A. Regattieri, and F. Sgarbossa (2013). Design and simulation of assembly line feeding systems in the automotive sector using supermarket, kanbans and tow trains: a general framework. *Journal of Management Control 24*(2), 187–208.

Fantini, P., M. Pinzone, and M. Taisch (2020). Placing the operator at the centre of industry 4.0 design: Modelling and assessing human activities within cyber-physical systems. *Computers & Industrial Engineering 139*, 105058.

Fast-Berglund, Å., F. Palmkvist, P. Nyqvist, S. Ekered, and M. Åkerman (2016). Evaluating cobots for final assembly. *Procedia CIRP 44*, 175–180.

Fathi, M., A. Nourmohammadi, M. Ghobakhloo, and M. Yousefi (2020). Production sustainability via supermarket location optimization in assembly lines. *Sustainability 12*(11), 4728.

Fathi, M., V. Rodríguez, D. B. M. M. Fontes, and M. J. Alvarez (2015). A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines. *International Journal of Production Research 54*(3), 878–893.

Felsberger, A., B. Oberegger, and G. Reiner (2016). A review of decision support systems for manufacturing systems. In *Proceedings of the 1st International Workshop on Science, Application and Methods in Industry 4.0*. October 18th-19th, Graz, Austria.

Ferrer, A. J., D. G. Pérez, and R. S. González (2016). Multi-cloud platform-as-a-service model, functionalities and approaches. *Procedia Computer Science 97*, 63–72.

Ferrer, B. R., U. Muhammad, W. Mohammed, and J. M. Lastra (2018). Implementing and visualizing ISO 22400 key performance indicators for monitoring discrete manufacturing systems. *Machines 6*(3), 39.

Fisher, R. A. (1935). *The design of experiments* (1st ed.)., Chapter 1, pp. 1–12. Edinburgh: Tweeddale Court, London: 33 Paternoster Row, E.C: Oliver and Boyd.

Flores-García, E., Y. Jeong, and M. Wiktorsson (2021). Applying machine learning for adaptive scheduling and execution of material handling in smart production logistics. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Eds.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Volume 631 of *IFIP Advances in Information and Communication Technology*, pp. 28–36. Springer International Publishing.

Foidl, H. and M. Felderer (2015). Research challenges of industry 4.0 for quality management. In *International conference on enterprise resource planning systems*, pp. 121–137. Springer.

Forbus, K. D. (2008). Qualitative modeling. In *Handbook of Knowledge Representation*, Chapter 9, pp. 361–393. Elsevier.

Fragapane, G., R. De Koster, F. Sgarbossa, and J. O. Strandhagen (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research 294*(2), 405–426.

Fragapane, G., D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen (2020). Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Annals of Operations Research 0*(0), 1–19.

Fransoo, J. C. and W. G. M. M. Rutten (1994). A typology of production control situations in process industries. *International Journal of Operations & Production Management 14*(12), 47–57.

Gagliardi, J.-P., J. Renaud, and A. Ruiz (2012). Models for automated storage and retrieval systems: a literature review. *International Journal of Production Research 50*(24), 7110–7125.

Gao, J. Z., L. Prakash, and R. Jagatesan (2007). Understanding 2D-BarCode technology and applications in M-commerce - design and implementation of a 2D barcode processing solution. In *31st Annual International Computer Software and Applications Conference - Vol. 2 - (COMPSAC 2007)*. IEEE.

Garrido, J. M. (2002). Basic probability theory and random numbers. In *Performance Modeling of Operating Systems Using Object-Oriented Simulation*, Chapter 3, pp. 25–39. Springer, Boston, MA.

Georgakopoulos, D., P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan (2016). Internet of things and edge cloud computing roadmap for manufacturing. *IEEE Cloud Computing 3*(4), 66–73.

Ghalehkhondabi, I., D. Sormaz, and G. Weckman (2016). Multiple customer order decoupling points within a hybrid MTS/MTO manufacturing supply chain with uncertain demands in two consecutive echelons. *OPSEARCH 53*(4), 976–997.

Ghanei, S. and T. AlGeddawy (2016). A new model for sustainable changeability and production planning. *Procedia CIRP 57*, 522–526.

Ghasemi, A., R. Azzouz, G. Laipple, K. E. Kabak, and C. Heavey (2020). Optimizing capacity allocation in semiconductor manufacturing photolithography area – case study: Robert bosch. *Journal of Manufacturing Systems 54*, 123–137.

Gibson, I., D. W. Rosen, and B. Stucker (2010). Introduction and basic principles. In *Additive Manufacturing Technologies*, pp. 20–35. Springer US.

Gits, C. W. (1994). Structuring maintenance control systems. *International Journal of Operations & Production Management 14*(7), 5–17.

Gligor, D. M., M. C. Holcomb, and T. P. Stank (2013). A multidisciplinary approach to supply chain agility: Conceptualization and scale development. *Journal of Business Logistics 34*(2), 94–108.

Glock, C. H. and E. H. Grosse (2012). Storage policies and order picking strategies in u-shaped order-picking systems with a movable base. *International Journal of Production Research 50*(16), 4344–4357.

Glock, C. H., E. H. Grosse, H. Abedinnia, and S. Emde (2019). An integrated model to improve ergonomic and economic performance in order picking by rotating pallets. *European Journal of Operational Research 273*(2), 516–534.

Godina, R., E. Rodrigues, and J. Matias (2017). An alternative test of normality for improving SPC in a portuguese automotive SME. In *Lecture Notes in Management and Industrial Engineering*, pp. 277–285. Springer International Publishing.

Goh, M., O. Jihong, and T. Chung-Piaw (2001). Warehouse sizing to minimize inventory and storage costs. *Naval Research Logistics 48*(4), 299–312.

Golz, J., R. Gujjula, H.-O. Günther, S. Rinderer, and M. Ziegler (2012). Part feeding at high-variant mixed-model assembly lines. *Flexible Services and Manufacturing Journal 24*(2), 119–141.

Gonçalves Machado, C., M. P. Winroth, and E. H. D. Ribeiro da Silva (2019). Sustainable manufacturing in industry 4.0: an emerging research agenda. *International Journal of Production Research 58*(5), 1462–1484.

Gosselin, M. (2006). A review of activity-based costing: Technique, implementation, and consequences. In C. S. Chapman, A. G. Hopwood, and M. D. Shields (Eds.), *Handbooks of Management Accounting Research*, Volume 2, pp. 641–671. Elsevier.

Gowrishankar, K., C. Rajendran, and G. Srinivasan (2001). Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *European Journal of Operational Research 132*(3), 643–665.

Grall, A., C. Bérenguer, and L. Dieulle (2002). A condition-based maintenance policy for stochastically deteriorating systems. *Reliability Engineering & System Safety 76*(2), 167–180.

Grangeon, N., P. Leclaire, and S. Norre (2011). Heuristics for the re-balancing of a vehicle assembly line. *International Journal of Production Research 49*(22), 6609–6628.

Grassi, A., G. Guizzi, L. C. Santillo, and S. Vespoli (2020). Assessing the performances of a novel decentralised scheduling approach in industry 4.0 and cloud manufacturing contexts. *International Journal of Production Research 59*(20), 6034–6053.

Gray, A. E., U. S. Karmarkar, and A. Seidmann (1992). Design and operation of an order-consolidation warehouse: Models and application. *European Journal of Operational Research 58*(1), 14–36.

Groover, M. P. (2016). *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India.

Gu, J., M. Goetschalckx, and L. F. McGinnis (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research 177*(1), 1–21.

Gu, J., M. Goetschalckx, and L. F. McGinnis (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research 203*(3), 539–549.

Gudanowska, A. E., J. P. Alonso, and A. Törmänen (2018). What competencies are needed in the production industry? the case of the podlaskie region. *Engineering Management in Production and Services 10*(1), 65–74.

Gue, K. R. and R. D. Meller (2009). Aisle configurations for unit-load warehouses. *IIE Transactions 41*(3), 171–182.

Guo, S., T.-M. Choi, B. Shen, and S. Jung (2019). Inventory management in mass customization operations: A review. *IEEE Transactions on Engineering Management 66*(3), 412–428.

Hamzaoui, M. A., T. Arbaoui, Z. Sari, and F. Yalaoui (2019). Exact and heuristic approaches for the design of automated storage and retrieval systems (AS/RS). In *2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*. IEEE.

Hausman, W. H., L. B. Schwarz, and S. C. Graves (1976). Optimal storage assignment in automatic warehousing systems. *Management Science 22*(6), 629–638.

Henn, S., S. Koch, and G. Wäscher (2012). Order batching in order picking warehouses: A survey of solution approaches. In *Warehousing in the Global Supply Chain*, pp. 105–137. Springer London.

Heragu, S. S., L. Du, R. J. Mantel, and P. C. Schuur (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research 43*(2), 327–338.

Hermann, M., T. Pentek, and B. Otto (2016). Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii international conference on system sciences (HICSS)*, pp. 3928–3937. IEEE.

Herrmann, C., C. Schmidt, D. Kurle, S. Blume, and S. Thiede (2014). Sustainability in manufacturing and factories of the future. *International Journal of Precision Engineering and Manufacturing-Green Technology 1*(4), 283–292.

Ho, W. R., O. Maghazei, and T. Netland (2022). Manufacturing repurposing: A literature review. In *IFIP Advances in Information and Communication Technology*, pp. 160–172. Springer Nature Switzerland.

Hon, K. K. B. (2005). Performance and evaluation of manufacturing systems. *CIRP Annals 54*(2), 139–154.

Hong, S., A. L. Johnson, and B. A. Peters (2012). Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research 221*(3), 557–570.

Hong, S., A. L. Johnson, and B. A. Peters (2015). Quantifying picker blocking in a bucket brigade order picking system. *International Journal of Production Economics 170*, 862–873.

Horňáková, N., L. Jurík, H. H. Chovanová, D. Cagáňová, and D. Babčanová (2019). AHP method application in selection of appropriate material handling equipment in selected industrial enterprise. *Wireless Networks 27*(3), 1683–1691.

Hosseini-Nasab, H., S. Fereidouni, S. M. T. Fatemi Ghomi, and M. B. Fakhrzad (2017). Classification of facility layout problems: a review study. *The International Journal of Advanced Manufacturing Technology 94*(1-4), 957–977.

Huang, S., B. Wang, X. Li, P. Zheng, D. Mourtzis, and L. Wang (2022). Industry 5.0 and society 5.0—comparison, complementation and co-evolution. *Journal of Manufacturing Systems 64*, 424–428.

IEC (2015). Enterprise-control system integratio - part 4: Objects and attributes for manufacturing operations management integration. URL: https://www.iso.org/standard/63740.html.

Ingalls, R. G. (2001). Introduction to simulation. In *Proceedings of the Winter Simulation Conference*. IEEE.

ISO (2005). Information technology - Open Distributed Processing - Unified Modeling Language (UML) version 1.4.2. URL: https://www.iso.org/standard/32620.html.

ISO (2014a). Automation systems and integration - key performance indicators (kpis) for manufacturing operations management - part 1: Overview, concepts and terminology.

ISO (2014b). Automation systems and integration - key performance indicators (kpis) for manufacturing operations management - part 2: Definitions and descriptions.

ISO (2019). Enterprise modelling and architecture - requirements for enterprise-referencing architectures and methodologies. URL: https://www.iso.org/standard/71890.html.

ISO (2020). Enterprise modelling and architecture - constructs for enterprise modelling. URL: https://www.iso.org/standard/74491.html.

Jahangirian, M., T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research 203*(1), 1–13.

Jiang, J.-R. (2018). An improved cyber-physical systems architecture for industry 4.0 smart factories. *Advances in Mechanical Engineering 10*(6), 168781401878419.

Jørgensen, H. D. (2009). Enterprise modeling – what we have learned, and what we have not. In *Lecture Notes in Business Information Processing*, pp. 3–7. Springer Berlin Heidelberg.

Jovane, F., H. Yoshikawa, L. Alting, C. R. Boër, E. Westkamper, D. Williams, M. Tseng, G. Seliger, and A. M. Paci (2008). The incoming global technological and industrial revolution towards competitive sustainable manufacturing. *CIRP Annals 57*(2), 641–659.

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research 4*, 237–285.

Kang, H. S., S. D. Noh, J. Y. Son, H. Kim, H. J. Park, and J. Y. Lee (2018). The FaaS system using additive manufacturing for personalized production. *Rapid Prototyping Journal 24*(9), 1486–1499.

Kaplan, R. S. (2009). Conceptual foundations of the balanced scorecard. In C. S. Chapman, A. G. Hopwood, and M. Shields (Eds.), *Handbooks of Management Accounting Research*, Volume 3, pp. 1253–1269. Elsevier.

Kateel, G. P., M. Kamath, and D. B. Pratt (1996). An overview of CIM enterprise modeling methodologies. In *Proceedings of the 28th conference on Winter simulation - WSC '96*. ACM Press.

Kaur, R., A. Awasthi, and K. Grzybowska (2019). Evaluation of key skills supporting industry 4.0—a review of literature and practice. In *Sustainable Logistics and Production in Industry 4.0*, pp. 19–29. Springer International Publishing.

Keen, P. G. W. (1980). Decision support systems: a research perspective. In G. Fick and J. Ralph H. Sprague (Eds.), *Decision support systems: Issues and challenges: Proceedings of an international task force meeting*, Volume 11, pp. 23–44.

Keen, P. G. W. and M. S. S. Morton (1978). *Decision Support Systems: An Organizational Perspective*, Chapter 1, pp. 1–32. ADDISON-WESLEY.

Keung, K. L., C. K. M. Lee, and P. Ji (2021). Data-driven order correlation pattern and storage location assignment in robotic mobile fulfillment and process automation system. *Advanced Engineering Informatics 50*, 101369.

Khodke, P. M. and A. S. Bhongade (2013). Real-time scheduling in manufacturing system with machining and assembly operations: a state of art. *International Journal of Production Research 51*(16), 4966–4978.

Kilic, H. S. and M. B. Durmusoglu (2015). Advances in assembly line parts feeding policies: a literature review. *Assembly Automation 35*(1), 57–68.

Klenk, E. and S. Galka (2019). Analysis of real-time tour building and scheduling strategies for in-plant milk-run systems with volatile transportation demand. *IFAC-PapersOnLine 52*(13), 2110–2115.

Ko, J., S. J. Hu, and T. Huang (2005). Reusability assessment for manufacturing systems. *CIRP Annals 54*(1), 113–116.

Kochhar, J. S. and S. S. Heragu (1999). Facility layout design in a changing environment. *International Journal of Production Research 37*(11), 2429–2446.

Kochikar, V. P. and T. T. Narendran (1994). On using abstract models for analysis of flexible manufacturing systems. *International Journal of Production Research 32*(10), 2303–2322.

Kong, X. T. R., H. Luo, G. Q. Huang, and X. Yang (2018). Industrial wearable system: the human-centric empowering technology in industry 4.0. *Journal of Intelligent Manufacturing 30*(8), 2853–2869.

Koren, Y. (2006). General RMS characteristics. Comparison with dedicated and flexible systems. In *Reconfigurable manufacturing systems and transformable factories*, pp. 27–45. Springer.

Koren, Y. (2010). *The Global Manufacturing Revolution.* John Wiley & Sons, Inc.

Koren, Y., X. Gu, and W. Guo (2018). Choosing the system configuration for high-volume manufacturing. *International Journal of Production Research 56*(1-2), 476–490.

Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel (1999). Reconfigurable manufacturing systems. *CIRP Annals 48*(2), 527–540.

Koren, Y., W. Wang, and X. Gu (2016). Value creation through design for scalability of reconfigurable manufacturing systems. *International Journal of Production Research 55*(5), 1227–1242.

Kosanke, K., F. Vernadat, and M. Zelm (1999). CIMOSA: enterprise engineering and integration. *Computers in industry 40*(2-3), 83–97.

Kovács, G. and S. Kot (2017). Facility layout redesign for efficiency improvement and cost reduction. *Journal of Applied Mathematics and Computational Mechanics 16*(1), 63–74.

Kovalev, S., I. Chalamon, and S. E. Collignon (2019). Minimizing maximum job dependent ergonomic risk. *International Journal of Production Research 59*(1), 95–101.

Kübler, P., C. H. Glock, and T. Bauernhansl (2020). A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses. *Computers & Industrial Engineering 147*, 106645.

Kulak, O., Y. Sahin, and M. E. Taner (2011). Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. *Flexible Services and Manufacturing Journal 24*(1), 52–80.

Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21 st century. *Journal of Intelligent Manufacturing 18*(2), 273–284.

Labrousse, M. and A. Bernard (2008). FBS-PPRE, an enterprise knowledge lifecycle model. In *Methods and Tools for Effective Knowledge Life-Cycle-Management*, pp. 285–305. Springer Berlin Heidelberg.

Ladier, A.-L., A. G. Greenwood, G. Alpan, and H. Hales (2014). Issues in the complementary use of simulation and optimization modeling. Research report, –. Cahiers Leibnitz n°211.

Laguna, M. (1999). A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times. *IIE Transactions 31*(2), 125–134.

Larson, T. N., H. March, and A. Kusiak (1997). A heuristic approach to warehouse layout with class-based storage. *IIE Transactions 29*(4), 337–348.

Lasi, H., P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann (2014). Industry 4.0. *Business & Information Systems Engineering 6*(4), 239–242.

Le-Duc, T. and M. De Koster (2005). Determining number of zones in a pick-and-pack orderpicking system. Technical report, ERASMUS Research Institute OF Management.

Lee, C. K. M., B. Lin, K. K. H. Ng, Y. Lv, and W. C. Tai (2019). Smart robotic mobile fulfillment system with dynamic conflict-free strategies considering cyber-physical integration. *Advanced Engineering Informatics 42*, 100998.

Lee, C.-Y. (2020). A continuous review inventory model with complex correlations among components. *Journal of Intelligent & Fuzzy Systems 39*(5), 6935–6947.

Lee, H.-H. (2005). A cost/benefit model for investments in inventory and preventive maintenance in an imperfect production system. *Computers & Industrial Engineering 48*(1), 55–68.

Lee, H. L. and M. J. Rosenblatt (1987). Simultaneous determination of production cycle and inspection schedules in a production system. *Management Science 33*(9), 1125–1136.

Lee, J., B. Bagheri, and H.-A. Kao (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters 3*, 18–23.

Lee, J., H. Davari, J. Singh, and V. Pandhare (2018). Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manufacturing Letters 18*, 20–23.

Lemieux, A.-A., S. Lamouri, R. Pellerin, and S. Tamayo (2015). Development of a leagile transformation methodology for product development. *Business Process Management Journal 21*(4), 791–819.

Leung, K. H., C. K. M. Lee, and K. L. Choy (2020). An integrated online pick-to-sort order batching approach for managing frequent arrivals of b2b e-commerce orders under both fixed and variable time-window batching. *Advanced Engineering Informatics 45*, 101125.

Li, L., Y. Wang, and K.-Y. Lin (2020). Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization. *Journal of Intelligent Manufacturing 32*(2), 545–558.

Li, Z., H. Shen, H. Li, G. Xia, P. Gamba, and L. Zhang (2017). Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. *Remote Sensing of Environment 191*, 342–358.

Liao, Y., E. D. F. R. Loures, and F. Deschamps (2018). Industrial internet of things: A systematic literature review and insights. *IEEE Internet of Things Journal 5*(6), 4515–4525.

Lindemann, C., U. Jahnke, M. Moi, and R. Koch (2012). Analyzing product lifecycle costs for a better understanding of cost drivers in additive manufacturing. In *23rd Annual International Solid Freeform Fabrication Symposium: An Additive Manufacturing Conference, Proceedings, August 6-8, 2012*, Solid freeform fabrication proceedings.

Lins, T., R. A. R. Oliveira, L. H. A. Correia, and J. Sa Silva (2018). Industry 4.0 retrofitting. In *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*. IEEE.

Liu, Y. and X. Xu (2016). Industry 4.0 and cloud manufacturing: A comparative analysis. *Journal of Manufacturing Science and Engineering 139*(3), 8.

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration 6*, 1–10.

Lu, Y., C. Liu, K. I.-K. Wang, H. Huang, and X. Xu (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing 61*, 101837.

Ma, L., D. Chablat, F. Bennis, W. Zhang, and F. Guillaume (2010). A new muscle fatigue and recovery model and its ergonomics application in human simulation. *Virtual and Physical Prototyping 5*(3), 123–137.

Maguire, K. (2015). Lean and IT—working together? an exploratory study of the potential conflicts between lean thinking and the use of information technology in organisations today. In *Measuring Operations Performance*, pp. 31–60. Springer International Publishing.

Makssoud, F., O. Battaïa, A. Dolgui, K. Mpofu, and O. Olabanji (2015). Re-balancing problem for assembly lines: new mathematical model and exact solution method. *Assembly Automation 35*(1), 16–21.

Malakuti, S., P. V. Schalkwyk, B. Boss, C. R. Sastry, V. Runkana, S.-W. Lin, S. Rix, G. Green, K. Baechle, and S. V. Nath (2020). Digital twins for industrial applications. definition, business values, design aspects, standards and use cases. Technical report, Industrial Internet Consortium.

Malinowska, M., A. Rzeczycki, and M. Sowa (2018). Roadmap to sustainable warehouse. *SHS Web of Conferences 57*, 01028.

Manceaux, A., H. Bril El-Haouzi, W. Ramdane Cherif-Khettaf, and M. L. Bentaha (2016). Iterated local search for dynamic assembly line rebalancing problem. *IFAC-PapersOnLine 49*(12), 515–519.

Manvi, S. S. and G. K. Shyam (2014). Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications 41*, 424–440.

Maravelias, C. T. and C. Sung (2009). Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering 33*(12), 1919–1930.

Martínez, K. P., R. Morabito, and E. A. V. Toso (2018). A coupled process configuration, lot-sizing and scheduling model for production planning in the molded pulp industry. *International Journal of Production Economics 204*, 227–243.

Martínez-Costa, C., M. Mas-Machuca, E. Benedito, and A. Corominas (2014). A review of mathematical programming models for strategic capacity planning in manufacturing. *International Journal of Production Economics 153*, 66–85.

Masae, M., C. H. Glock, and E. H. Grosse (2020). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics 224*, 107564.

Mason-Jones, R., B. Naylor, and D. R. Towill (2000). Lean, agile or leagile? matching your supply chain to the marketplace. *International Journal of Production Research 38*(17), 4061–4070.

Mayer, R. J., M. K. Painter, and P. S. de Witte (1994). *IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications*. Knowledge Based Systems.

McCarthy, I. and M. Menicou (2002). A classification schema of manufacturing decisions for the GRAI enterprise modelling technique. *Computers in Industry 47*(3), 339–355.

Melacini, M., S. Perotti, and A. Tumino (2010). Development of a framework for pick-and-pass order picking system design. *The International Journal of Advanced Manufacturing Technology 53*(9-12), 841–854.

Meng, G., S. S. Heragu *, and H. Zijm (2004). Reconfigurable layout problem. *International Journal of Production Research 42*(22), 4709–4729.

Meng, K., Q. Tang, Z. Zhang, and C. Yu (2021). Solving multi-objective model of assembly line balancing considering preventive maintenance scenarios using heuristic and grey wolf optimizer algorithm. *Engineering Applications of Artificial Intelligence 100*, 104183.

Millet, P.-A., L. Trilling, T. Moyaux, and O. Sakka (2013). Ontology of SCOR for the strategic alignment of organizations and information systems. In V. Botta-Genoulaz, J.-P. Campagne, D. Llerena, and C. Pellegrin (Eds.), *Supply Chain Performance: Collaboration, Alignment and Coordination*, pp. 171–210. London: John Wiley & Sons, Inc.

Minciu, M., F.-A. Berar, and R. C. Dobrea (2020). New decision systems in the VUCA world. *Management & Marketing 15*(2), 236–254.

Mincsovics, G., T. Tan, and O. Alp (2009). Integrated capacity and inventory management with capacity acquisition lead times. *European Journal of Operational Research 196*(3), 949–958.

Mintzberg, H., D. Raisinghani, and A. Theoret (1976). The structure of "unstructured" decision processes. *Administrative science quarterly 21*(2), 246–275.

Mitra, R., M. Shukla, A. Goswami, and M. K. Tiwari (2021). Digitization of real-time predictive maintenance for high speed machine equipment. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Eds.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Volume 631 of *IFIP Advances in Information and Communication Technology*, pp. 132–140. Springer International Publishing.

Mohsen, M. D. H. (2010). A framework for selection of material handling equipment in manufacturing and logistics facilities. *Journal of Manufacturing Technology Management 21*(2), 246–268.

Moldavska, A. and T. Welo (2017). The concept of sustainable manufacturing and its definitions: A content-analysis based literature review. *Journal of Cleaner Production 166*, 744–755.

Montanari, R., R. Micale, E. Bottani, A. Volpi, and G. La Scalia (2021). Evaluation of routing policies using an interval-valued TOPSIS approach for the allocation rules. *Computers & Industrial Engineering 156*, 107256.

Moreira, M. C. O., R. Pastor, A. M. Costa, and C. Miralles (2017). The multi-objective assembly line worker integration and balancing problem of type-2. *Computers & Operations Research 82*, 114–125.

Morgan, J., M. Halton, Y. Qiao, and J. G. Breslin (2021). Industry 4.0 smart reconfigurable manufacturing machines. *Journal of Manufacturing Systems 59*, 481–506.

Mosavi, A., S. Ardabili, and A. R. Várkonyi-Kóczy (2020). List of deep learning models. In *Lecture Notes in Networks and Systems*, pp. 202–214. Springer International Publishing.

Müller, J. M., D. Kiel, and K.-I. Voigt (2018). What drives the implementation of industry 4.0? the role of opportunities and challenges in the context of sustainability. *Sustainability 10*(1), 247.

Mura, M. D. and G. Dini (2019). Designing assembly lines with humans and collaborative robots: A genetic approach. *CIRP Annals 68*(1), 1–4.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nagy, L., T. Ruppert, and J. Abonyi (2020). Analytic hierarchy process and multilayer network-based method for assembly line balancing. *Applied Sciences 10*(11), 3932.

Nahmias, S. and T. L. Olsen (2020). *Production and Operations Analytics*. Waveland press, Incorporated.

Naylor, J. B., M. M. Naim, and D. Berry (1999). Leagility: Integrating the lean and agile manufacturing paradigms in the total supply chain. *International Journal of Production Economics 62*(1-2), 107–118.

Neely, A., M. Gregory, and K. Platts (1995). Performance measurement system design: a literature review and research agenda. *International Journal of Operations & Production Management 15*(4), 80–116.

Negahban, A. and J. S. Smith (2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems 33*(2), 241–261.

Nestman, C. H. and J. C. Windsor (1985). Decision support systems: A perspective for industrial engineers. *IIE Transactions 17*(1), 38–46.

Neumann, W. P., S. Winkelhaus, E. H. Grosse, and C. H. Glock (2021). Industry 4.0 and the human factor – a systems framework and analysis methodology for successful development. *International Journal of Production Economics 233*, 107992.

Nian, R., J. Liu, and B. Huang (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering 139*, 106886.

Niroomand, I., O. Kuzgunkaya, and A. A. Bulgak (2012). Impact of reconfiguration characteristics for capacity investment strategies in manufacturing systems. *International Journal of Production Economics 139*(1), 288–301.

Noran, O. (2004). UML vs. IDEF: AN ONTOLOGY-ORIENTED COMPARATIVE STUDY IN VIEW OF BUSINESS MODELLING. In *Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems*. SciTePress - Science and and Technology Publications.

Nourmohammadi, A., H. Eskandari, and M. Fathi (2018). Design of stochastic assembly lines considering line balancing and part feeding with supermarkets. *Engineering Optimization 51*(1), 63–83.

Olhager, J. (2010). The role of the customer order decoupling point in production and supply chain management. *Computers in Industry 61*(9), 863–868.

Olmos, J., R. Florencia, V. García, M. V. González, G. Rivera, and P. Sánchez-Solís (2021). Metaheuristics for order picking optimisation: A comparison among three swarm-intelligence algorithms. In *Technological and Industrial Applications Associated With Industry 4.0*, pp. 177–194. Springer International Publishing.

Osterwalder, A. and Y. Pigneur (2010). *Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons.

Ostic, J. K. and C. E. Cannon (1996). An introduction to enterprise modeling and simulation. Technical report, Los Alamos National Laboratory,University Of California.

Otto, A. and O. Battaïa (2017). Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: A survey. *Computers & Industrial Engineering 111*, 467–480.

Özdemir, R. G. and Z. Ayağ (2010). An integrated approach to evaluating assembly-line design alternatives with equipment selection. *Production Planning & Control 22*(2), 194–206.

Parikh, P. J. and R. D. Meller (2008). Selecting between batch and zone order picking strategies in a distribution center. *Transportation Research Part E: Logistics and Transportation Review 44*(5), 696–719.

Park, B. C., E. H. Frazelle, and J. A. White (1999). Buffer sizing models for end-of-aisle order picking systems. *IIE Transactions 31*(1), 31–38.

Park, K. T., Y. W. Nam, H. S. Lee, S. J. Im, S. D. Noh, J. S. Son, and H. Kim (2019). Design and implementation of a digital twin application for a connected micro smart factory. *International Journal of Computer Integrated Manufacturing 32*(6), 596–614.

Pattanaik, L. N. and A. Jena (2020). Selection of optimal assembly strategy at an engine assembly plant using fuzzy multi-criteria decision-making tools. *International Journal of Industrial and Systems Engineering 36*(2), 281.

Perera, S., G. Janakiraman, and S.-C. Niu (2017). Optimality of (s , s) inventory policies under renewal demand and general cost structures. *Production and Operations Management 27*(2), 368–383.

Peron, M., G. Fragapane, F. Sgarbossa, and M. Kay (2020). Digital facility layout planning. *Sustainability 12*(8), 3349.

Peters, B. A. and L. F. McGinnis (1999). Strategic configuration of fexible assembly systems a single period approximation. *IIE Transactions 31*(4), 379–390.

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Operations & Production Management 17*(11), 1098–1111.

Petersen II, C. G. (2009). An evaluation of order picking policies for mail order companies. *Production and Operations Management 9*(4), 319–335.

Phadermrod, B., R. M. Crowder, and G. B. Wills (2019). Importance-performance analysis based swot analysis. *International Journal of Information Management 44*, 194–203.

Pick, R. A. (2008). Benefits of decision support systems. In *Handbook on Decision Support Systems 1*, pp. 719–730. Springer Berlin Heidelberg.

Pine, B. J. (1993). Mass customisation: The new frontier in business competition. harvard business school press, boston. *Australian Journal of Management 17*(2), 271–283.

Pinzone, M., P. Fantini, S. Perini, S. Garavaglia, M. Taisch, and G. Miragliotta (2017). Jobs and skills in industry 4.0: An exploratory research. In H. Lödding, R. Riedel, K.-D. Thoben, G. von Cieminski, and D. Kiritsis (Eds.), *Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing*, Cham, pp. 282–288. Springer International Publishing.

Pisching, M. A., M. A. O. Pessoa, F. Junqueira, D. J. d. S. Filho, and P. E. Miyagi (2018). An architecture based on RAMI 4.0 to discover equipment to process operations required by products. *Computers & Industrial Engineering 125*, 574–591.

Plehn, J., R. ZÃ¼st, F. Kimura, A. Sproedt, and P. SchÃ¶nsleben (2012). A method for determining a functional unit to measure environmental performance in manufacturing systems. *CIRP Annals 61*(1), 415–418.

Posadas, S., K. M. Kruger, C. M. Beazley, R. S. Salley, J. A. Stephenson, E. C. Thron, and J. D. Ward (2020). Forecasting parts demand using service data and machine learning volume 1. Technical report, LMI Tysons Corner United States.

Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research 275*(3), 795–821.

Prabaharan, T., P. R. Nakkeeran, and N. Jawahar (2006). Sequencing and scheduling of job and tool in a flexible manufacturing cell. *The International Journal of Advanced Manufacturing Technology 29*(7-8), 729–745.

Pragana, J. P. M., R. F. V. Sampaio, I. M. F. Bragança, C. M. A. Silva, and P. A. F. Martins (2021). Hybrid metal additive manufacturing: A state–of–the-art review. *Advances in Industrial and Manufacturing Engineering 2*, 100032.

Prifti, L., M. Knigge, H. Kienegger, and H. Krcmar (2017). A competency model for "industrie 4.0" employees. In J. M. Leimeister and W. Brenner (Eds.), *13th International Conference on Wirtschaftsinformatik (WI)*, pp. 46–60. University of St. Gallen.

Puthenveetil, S. C., C. P. Daphalapurkar, W. Zhu, M. C. Leu, X. F. Liu, J. K. Gilpin-Mcminn, and S. D. Snodgrass (2015). Computer-automated ergonomic analysis based on motion capture and assembly simulation. *Virtual Reality 19*(2), 119–128.

Radziwon, A., A. Bilberg, M. Bogers, and E. S. Madsen (2014). The smart factory: Exploring adaptive and flexible manufacturing solutions. *Procedia Engineering 69*, 1184–1190.

Rafiei, H., M. Rabbani, and M. Alimardani (2013). Novel bi-level hierarchical production planning in hybrid MTS/MTO production contexts. *International Journal of Production Research 51*(5), 1331–1346.

Raghuram, P. and A. Singh (2020). Warehouse optimisation using demand data analytics - a case study-based approach. *International Journal of Business Information Systems 35*(4), 519.

Rai, R., M. K. Tiwari, D. Ivanov, and A. Dolgui (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research 59*(16), 4773–4778.

Ramasesh, R., S. Kulkarni, and M. Jayakumar (2001). Agility in manufacturing systems: an exploratory modeling framework and simulation. *Integrated Manufacturing Systems 12*(7), 534–548.

Ramezanian, R., M. Saidi-Mehrabad, and E. Teimoury (2013). A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments. *The International Journal of Advanced Manufacturing Technology 66*(1), 347–361.

Reisig, W. (1985). *Petri Nets: An Introduction*, Chapter 2, pp. 18–31. Springer Berlin Heidelberg.

Rizkya, I., K. Syahputri, R. M. Sari, A. Anizar, I. Siregar, and E. Ginting (2018). Comparison of periodic review policy and continuous review policy for the automotive industry inventory system. *IOP Conference Series: Materials Science and Engineering 288*, 012085.

Robinson, S. (2008). Conceptual modelling for simulation part i: definition and requirements. *Journal of the Operational Research Society 59*(3), 278–290.

Rodič, B. (2017). Industry 4.0 and the new simulation modelling paradigm. *Organizacija 50*(3), 193–207.

Rojas-Reyes, J. J., E. L. Solano-Charris, and J. R. Montoya-Torres (2019). The storage location assignment problem: A literature review. *International Journal of Industrial Engineering Computations 10*(2), 199–224.

Roodbergen, K. J. and I. F. A. Vis (2006). A model for warehouse layout. *IIE Transactions 38*(10), 799–811.

Roodbergen, K. J. and I. F. A. Vis (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research 194*(2), 343–362.

Roodbergen, K. J., I. F. A. Vis, and G. D. Taylor (2014). Simultaneous determination of warehouse layout and control policies. *International Journal of Production Research 53*(11), 3306–3326.

Ross, D. T. (1977). Structured analysis (SA): A language for communicating ideas. *IEEE Transactions on Software Engineering SE-3*(1), 16–34.

Rossit, D. A., F. Tohmé, and M. Frutos (2019). An industry 4.0 approach to assembly line resequencing. *The International Journal of Advanced Manufacturing Technology 105*(9), 3619–3630.

Rouwenhorst, B., B. Reuter, V. Stockrahm, G.-J. Van Houtum, R. J. Mantel, and W. H. M. Zijm (2000). Warehouse design and control: Framework and literature review. *European journal of operational research 122*(3), 515–533.

Roux, D. J., J. L. Nel, G. Cundill, P. O'Farrell, and C. Fabricius (2017). Transdisciplinary research for systemic change: who to learn with, what to learn about and how to learn. *Sustainability Science 12*(5), 711–726.

Rubrico, J. I. U., T. Higashi, H. Tamura, and J. Ota (2011). Online rescheduling of multiple picking agents for warehouse management. *Robotics and Computer-Integrated Manufacturing 27*(1), 62–71.

Saetta, S., L. Paolini, L. Tiacci, and T. Altiok (2012). A decomposition approach for the performance analysis of a serial multi-echelon supply chain. *International Journal of Production Research 50*(9), 2380–2395.

Sala, R., F. Pirola, G. Pezzotta, and M. Vernieri (2021). Improving maintenance service delivery through data and skill-based task allocation. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Eds.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Volume 631 of *IFIP Advances in Information and Communication Technology*, pp. 202–211. Springer International Publishing.

Sali, M. and E. Sahin (2016). Line feeding optimization for just in time assembly lines: An application to the automotive industry. *International Journal of Production Economics 174*, 54–67.

Sancı, E. and M. Azizoğlu (2017). Rebalancing the assembly lines: exact solution approaches. *International Journal of Production Research 55*(20), 5991–6010.

Saraswat, A., U. Venkatadri, and I. Castillo (2015). A framework for multi-objective facility layout design. *Computers & Industrial Engineering 90*, 167–176.

Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation 7*(1), 12–24.

Satoglu, S. I. and I. E. Sahin (2013). Design of a just-in-time periodic material supply system for the assembly lines and an application in electronics industry. *The International Journal of Advanced Manufacturing Technology 65*(1), 319–332.

Sawik, T. (2002). Monolithic vs. hierarchical balancing and scheduling of a flexible assembly line. *European Journal of Operational Research 143*(1), 115–124.

Schemeleva, K., X. Delorme, and A. Dolgui (2018). Evaluation of solution approaches for a stochastic lot-sizing and sequencing problem. *International Journal of Production Economics 199*, 179–192.

Schemeleva, K., X. Delorme, A. Dolgui, and F. Grimaud (2012). Multi-product sequencing and lot-sizing under uncertainties: A memetic algorithm. *Engineering Applications of Artificial Intelligence 25*(8), 1598–1610.

Schenone, M., G. Mangano, S. Grimaldi, and A. C. Cagliano (2020). An approach for computing AS/r systems travel times in a class-based storage configuration. *International Journal of Production Research 8*(1), 273–290.

Schiele, H. and R. Jan Torn (2020). Cyber-physical systems with autonomous machine-to-machine communication: Industry 4.0 and its particular potential for purchasing and supply management. *International Journal of Procurement Management 13*(4), 507.

Schleich, B., N. Anwer, L. Mathieu, and S. Wartzack (2017). Shaping the digital twin for design and production engineering. *CIRP Annals 66*(1), 141–144.

Schmid, N. A. and V. Limère (2019). A classification of tactical assembly line feeding problems. *International Journal of Production Research 57*(24), 7586–7609.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines*. Contributions to Management Science. Physica-Verlag HD.

Scholz, A., S. Henn, M. Stuhlmann, and G. Wäscher (2016). A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research 253*(1), 68–84.

Scholz, A. and G. Wäscher (2017). Order batching and picker routing in manual order picking systems: the benefits of integrated routing. *Central European Journal of Operations Research 25*(2), 491–520.

Schuhmacher, J. and V. Hummel (2016). Decentralized control of logistic processes in cyber-physical production systems at the example of ESB logistics learning factory. *Procedia CIRP 54*, 19–24.

Shafer, S. M., H. J. Smith, and J. C. Linder (2005). The power of business models. *Business Horizons 48*(3), 199–207.

Shafiee, M. (2015). Maintenance strategy selection problem: an MCDM overview. *Journal of Quality in Maintenance Engineering 21*(4), 378–402.

Shah, L. A., A. Etienne, A. Siadat, and F. Vernadat (2014). Decision-making in the manufacturing environment using a value-risk graph. *Journal of Intelligent Manufacturing 27*(3), 617–630.

Shah, R. and P. T. Ward (2007). Defining and developing measures of lean production. *Journal of Operations Management 25*(4), 785–805.

Shannon, R. E. (1998). Introduction to the art and science of simulation. In *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*. IEEE.

Shi, Y., X. Guo, and Y. Yu (2017). Dynamic warehouse size planning with demand forecast and contract flexibility. *International Journal of Production Research 56*(3), 1313–1325.

Shiue, Y.-R., K.-C. Lee, and C.-T. Su (2018). Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering 125*, 604–614.

Shivanand, H. K., M. M. Benal, and V. Koti (2006). *Flexible manufacturing system*. New Age International.

Sillitto, H., J. Martin, D. McKinney, R. Griego, D. Dori, D. Krob, P. Godfrey, E. Arnold, and S. Jackson (2019). Systems engineering and system definitions. In *INCOSE*.

Silva, A., L. C. Coelho, M. Darvish, and J. Renaud (2020). Integrating storage location and order picking problems in warehouse planning. *Transportation Research Part E: Logistics and Transportation Review 140*, 102003.

Silva, A., K. J. Roodbergen, L. C. Coelho, and M. Darvish (2021). Estimating optimal abc zone sizes in manual warehouses. Technical report, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport.

Silvestri, L., A. Forcina, V. Introna, A. Santolamazza, and V. Cesarotti (2020). Maintenance transformation through industry 4.0 technologies: A systematic literature review. *Computers in Industry 123*, 103335.

Simon, H. A. (1960). *The new science of management decision* (1 ed.), Volume 3, Chapter 1, pp. 1–8. Harper & Brothers.

Singha, K., J. Buddhakulsomsiri, and P. Parthanadee (2017). Mathematical model of (r,q) inventory policy under limited storage space for continuous and periodic review policies with backlog and lost sales. *Mathematical Problems in Engineering 2017*, 1–9.

Söderberg, R., K. Wärmefjord, J. S. Carlson, and L. Lindkvist (2017). Toward a digital twin for real-time geometry assurance in individualized production. *CIRP Annals 66*(1), 137–140.

Song, X., E. Poirson, Y. Ravaut, and F. Bennis (2021). Interactive design optimization of layout problems. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Eds.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, Volume 634 of *IFIP Advances in Information and Communication Technology*, pp. 387–395. Springer International Publishing.

Sortrakul, N., H. L. Nachtmann, and C. R. Cassady (2005). Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine. *Computers in Industry 56*(2), 161–168.

Sperandio, S. (2005). *Usage de la modélisation multi-vue d'entreprise pour la conduite des systèmes de production*. Ph. D. thesis, Université Sciences et Technologies-Bordeaux I.

Spezzano, G. and A. Vinci (2015). Pattern detection in cyber-physical systems. *Procedia Computer Science 52*, 1016–1021.

Stanley, M. D. (1987). *Future perfect*. Wiley.

Steier, G., S. Peukert, and G. Lanza (2021). An integrated approach for developing production strategy and digital business model in global production. In *Production at the Leading Edge of Technology*, pp. 666–674. Springer International Publishing.

Stock, T. and G. Seliger (2016). Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP 40*, 536–541. 13th Global Conference on Sustainable Manufacturing - Decoupling Growth from Resource Use.

Stützle, T. and R. Ruiz (2017). Iterated local search. In R. Martí, P. Panos, and M. G. C. Resende (Eds.), *Handbook of Heuristics*, pp. 1–27. Springer International Publishing.

Syan, C. S. and G. Ramsoobag (2019). Maintenance applications of multi-criteria optimization: A review. *Reliability Engineering & System Safety 190*, 106520.

Tao, F., J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui (2017). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology 94*(9-12), 3563–3576.

Tao, F., Q. Qi, L. Wang, and A. Y. C. Nee (2019). Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering 5*(4), 653–661.

Terzi, S. and S. Cavalieri (2004). Simulation in the supply chain context: a survey. *Computers in Industry 53*(1), 3–16.

Theorin, A., K. Bengtsson, J. Provost, M. Lieder, C. Johnsson, T. Lundholm, and B. Lennartson (2017). An event-driven manufacturing information system architecture for industry 4.0. *International Journal of Production Research 55*(5), 1297–1311.

Thomopoulos, N. T. (2014). *Assembly Line Planning and Control*. Springer International Publishing.

Toha, M., D. E. Prastyo, and A. Saptari (2019). A comparison of continuous and periodic review on inventory components of dump trucks. In *2019 International Conference on Sustainable Engineering and Creative Computing (ICSECC)*, pp. 364–368. IEEE.

Toktay, L. B. and R. Uzsoy (1998). A capacity allocation problem with integer side constraints. *European Journal of Operational Research 109*(1), 170–182.

Touzout, F. A. and L. Benyoucef (2018). Multi-objective sustainable process plan generation in a reconfigurable manufacturing environment: exact and adapted evolutionary approaches. *International Journal of Production Research 57*(8), 2531–2547.

Tu, M., M.-F. Yang, S.-L. Kao, F.-C. Lin, M.-H. Wu, and C.-K. Lin (2020). Using a heuristic multi-objective genetic algorithm to solve the storage assignment problem for CPS-based pick-and-pass system. *Enterprise Information Systems 0*(0), 1–22.

Turban, E., J. E. Aronson, T.-P. Liang, and R. V. MacCarthy (2005). *Decision support systems and intelligent systems* (7th ed.), Volume 4. Pearson Prentice-Hall Upper Saddle River, NJ, USA.

Turban, E., R. Sharda, D. Delen, J. E. Aronson, T.-P. Liang, and D. King (2011). *Decision support and business intelligence systems* (9th ed.)., Chapter 2, pp. 38–188. Prentice Hall PressOne, Lake Street Upper Saddle River, NJ, USA.

Udhayakumar, P. and S. Kumanan (2010). Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm. *The International Journal of Advanced Manufacturing Technology 50*(9-12), 1075–1084.

Uhlemann, T. H.-J., C. Lehmann, and R. Steinhilper (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia CIRP 61*, 335–340.

Ullah, S., G. Zailin, W. Baoxi, M. Jahanzeb, and H. Shiyang (2014). A survey on assembly lines and its types. *Frontiers of Mechanical Engineering 9*(2), 95–105.

Ulrich, F. (2012). Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. *Software & Systems Modeling 13*(3), 941–962.

Unver, H. O. (2012). An ISA-95-based manufacturing intelligence system in support of lean initiatives. *The International Journal of Advanced Manufacturing Technology 65*(5-8), 853–866.

Urrutia, E. D. G., R. Aggoune, and S. Dauzère-Pérès (2014). Solving the integrated lot-sizing and job-shop scheduling problem. *International Journal of Production Research 52*(17), 5236–5254.

Van Gils, T., K. Ramaekers, A. Caris, and M. Cools (2016). The use of time series forecasting in zone order picking systems to predict order pickers' workload. *International Journal of Production Research 55*(21), 6380–6393.

Van Gils, T., K. Ramaekers, A. Caris, and R. B. M. De Koster (2018). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research 267*(1), 1–15.

Van Mieghem, J. A. (2003). Commissioned paper: Capacity management, investment, and hedging: Review and recent developments. *Manufacturing & Service Operations Management 5*(4), 269–302.

Van Nieuwenhuyse, I. and R. B. M. De Koster (2009). Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics 121*(2), 654–664.

Venkitasubramony, R. and G. K. Adil (2018). Designing a block stacked warehouse for dynamic and stochastic product flow: a scenario-based robust approach. *International Journal of Production Research 57*(5), 1345–1365.

Vercellis, C. (1991). Multi-criteria models for capacity analysis and aggregate planning in manufacturing systems. *International Journal of Production Economics 23*(1-3), 261–272.

Vernadat, F. (2002). UEML: Towards a unified enterprise modelling language. *International Journal of Production Research 40*(17), 4309–4321.

Vernadat, F. (2014). Enterprise modeling in the context of enterprise engineering: State of the art and outlook. *International Journal of Production Management and Engineering 2*(2), 57–73.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H.

Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17*, 261–272.

Volpentesta, A. P. and A. M. Felicetti (2011). Competence mapping through analysing research papers of a scientific community. In *Technological Innovation for Sustainability*, pp. 33–44. Springer Berlin Heidelberg.

Wang, C. and P. Jiang (2016). Manifold learning based rescheduling decision mechanism for recessive disturbances in RFID-driven job shops. *Journal of Intelligent Manufacturing 29*(7), 1485–1500.

Wang, D. and C.-G. Xu (1997). Hybrid push pull production control strategy simulation and its applications. *Production Planning & Control 8*(2), 142–151.

Wang, K., S. Chen, Z. Jiang, W. Zhou, and N. Geng (2021). Capacity allocation of an integrated production and service system. *Production and Operations Management 30*(8), 2765–2781.

Wang, K.-J. and P. H. Nguyen (2017). Capacity planning with technology replacement by stochastic dynamic programming. *European Journal of Operational Research 260*(2), 739–750.

Wang, S., J. Wan, D. Li, and C. Zhang (2016). Implementing smart factory of industrie 4.0: An outlook. *International Journal of Distributed Sensor Networks 12*(1), 3159805.

Wang, S., J. Wan, D. Zhang, D. Li, and C. Zhang (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks 101*, 158–168.

Wang, X., S. K. Ong, and A. Y. C. Nee (2016). A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing 4*(1), 1–22.

Watkins, C. J. C. H. and P. Dayan (1992). Q-learning. *Machine Learning 8*(3-4), 279–292.

Weber, E. and O. Coskunoglu (1990). Descriptive and prescriptive models of decision-making: implications for the development of decision aids. *IEEE Transactions on Systems, Man, and Cybernetics 20*(2), 310–317.

Weinstein, L. and C.-H. Chung (1999). Integrating maintenance and production decisions in a hierarchical production planning environment. *Computers & Operations Research 26*(10-11), 1059–1074.

White, J. A. and R. L. Francis (1971). Normative models for some warehouse sizing problems. *A I I E Transactions 3*(3), 185–190.

Whitman, L. E., B. L. Huff, and A. R. Presley (1999). The needs and issues associated with representing and integrating multiple views of the enterprise. In *Information Infrastructure Systems for Manufacturing II*, pp. 139–152. Springer US.

Wilhelm, W. E. (1999). A column-generation approach for the assembly system design problem with tool changes. *International Journal of Flexible Manufacturing Systems 11*(2), 177–205.

Winkelhaus, S., E. H. Grosse, and S. Morana (2021). Towards a conceptualisation of order picking 4.0. *Computers & Industrial Engineering 159*, 107511.

Won, J. and S. Olafsson (2005). Joint order batching and order picking in warehouse operations. *International Journal of Production Research 43*(7), 1427–1442.

Wu, B. (1992). *Manufacturing systems design and analysis*, Chapter 1, pp. 3–27. Springer Science & Business Media.

Wu, C.-H., W.-C. Chen, and H.-C. Chiu (2021). Dynamic expansion of flexible capacity with and without pricing coordination. *Computers & Industrial Engineering 161*, 107610.

Wuest, T., D. Weimer, C. Irgens, and K.-D. Thoben (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research 4*(1), 23–45.

Xiao, Y., J. Chen, and C.-Y. Lee (2010). Optimal decisions for assemble-to-order systems with uncertain assembly capacity. *International Journal of Production Economics 123*(1), 155–165.

Xu, L. D. and L. Duan (2018). Big data for cyber physical systems in industry 4.0: a survey. *Enterprise Information Systems 13*(2), 148–169.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing 28*(1), 75–86.

Xu, X., Y. Lu, B. Vogel-Heuser, and L. Wang (2021). Industry 4.0 and industry 5.0—inception, conception and perception. *Journal of Manufacturing Systems 61*, 530–535.

Yang, H., S. Kumara, S. T. S. Bukkapatnam, and F. Tsung (2019). The internet of things for smart manufacturing: A review. *IISE Transactions 51*(11), 1190–1216.

Yang, H. and M. Tate (2012). A descriptive literature review and classification of cloud computing research. *Communications of the Association for Information Systems 31*(1), 2.

Yang, J. H. and K. H. Kim (2006). A grouped storage method for minimizing relocations in block stacking systems. *Journal of Intelligent Manufacturing 17*(4), 453–463.

Yang, P., L. Miao, Z. Xue, and L. Qin (2013). An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems. *Journal of Intelligent Manufacturing 26*(6), 1145–1159.

Yin, S. and O. Kaynak (2015). Big data for modern industry: Challenges and trends [point of view]. *Proceedings of the IEEE 103*(2), 143–146.

Yu, C., W. Zhang, X. Xu, Y. Ji, and S. Yu (2015). Data mining based multi-level aggregate service planning for cloud manufacturing. *Journal of Intelligent Manufacturing 29*(6), 1351–1361.

Yu, M. and R. B. M. De Koster (2009). The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research 198*(2), 480–490.

Yuan, X.-M. (2020). Impact of industry 4.0 on inventory systems and optimization. In *Industry 4.0 - Impact on Intelligent Logistics and Manufacturing*. IntechOpen.

Yusuf, Y. Y., M. Sarhadi, and A. Gunasekaran (1999). Agile manufacturing:. *International Journal of Production Economics 62*(1-2), 33–43.

Zacharewicz, G., A. Pirayesh-Neghab, M. Seregni, Y. Ducq, and G. Doumeingts (2017). Simulation-based enterprise management. In *Simulation Foundations, Methods and Applications*, pp. 261–289. Springer International Publishing.

Zha, J. and J.-J. Yu (2014). A hybrid ant colony algorithm for u-line balancing and rebalancing in just-in-time production environment. *Journal of Manufacturing Systems 33*(1), 93–102.

Zheng, P., H. wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu (2018). Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering 13*(2), 137–150.

Zheng, Y., S. Qiu, F. Shen, and C. He (2020). RFID-based material delivery method for mixed-model automobile assembly. *Computers & Industrial Engineering 139*, 106023.

Zhong, R. Y., X. Xu, E. Klotz, and S. T. Newman (2017). Intelligent manufacturing in the context of industry 4.0: A review. *Engineering 3*(5), 616–630.

Zhou, B. and Z. Zhu (2020). A dynamic scheduling mechanism of part feeding for mixed-model assembly lines based on the modified neural network and knowledge base. *Soft Computing 25*(1), 291–319.

Zhou, B.-H. and C.-Y. Shen (2018). Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration. *Journal of Cleaner Production 192*, 293–305.

Zhou, J. and J. Shi (2008). RFID localization algorithms and applications—a review. *Journal of Intelligent Manufacturing 20*(6), 695–707.

Zhou, M. and F. DiCesare (1993). Petri nets and manufacturing systems. In *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Chapter 2, pp. 13–29. Springer US.

Zhuang, C., J. Liu, and H. Xiong (2018). Digital twin-based smart production management and control framework for the complex product assembly shop-floor. *The International Journal of Advanced Manufacturing Technology 96*(1-4), 1149–1163.

Zolghadri, M., T. Lecompte, and J. P. Bourrieres (2001). From enterprise modelling to decision support system definition for virtual enterprise. *IFAC Proceedings Volumes 34*(17), 69–74.

Appendix A

Publications

- El Alaoui El Abdellaoui, M., E. Bricard, F. Grimaud, P. Gianessi, and X. Delorme (2020). Scalable, reconfigurable simulation models in industry 4.0-oriented enterprise modeling. In 2020 Winter Simulation Conference (WSC). IEEE.
- El Alaoui El Abdellaoui, M., F. Grimaud, P. Gianessi, and X. Delorme (2019). Integrated decision process to design manufacturing systems towards industry 4.0. IFAC-PapersOnLine 52 (13), 1373-1378.

Appendix B

MS Features

This appendix presents the MS features discussed in Chapter 3. We use the PPRE-CT approach to represent all information in MS, as shown in the six tables below. Abbreviations and codes used in each table are defined in Table B.1. The code format is Class.Category.numberID, and we use the gray color only to distinguish categories.

Table B.1: Abbreviations and encoding classes and categories.

Classes	Code	Categories	Code	Words	Abbreviation
Process	P	General	g	Information	info
Product	Z	Demand	μ	Dependent	dep
Resource	R	Sets	Π	Independent	indep
Externel Effect	E	Configuration	β	Unit(s)	u
Cost	C	Investment	γ	Unit time	ut
Time	T	Orders	θ	In period t	(t)
		Inventory	α	Planning horizon	PH
		Production	p	Year(s)	y
		Supply	s	Day(s)	d
		Storage	w	Hour(s)	h
		Distribution	δ	Minute(s)	mn
		Constraints	c	Number	#
		Assumptions	a	Configuration(s)	config
		Objectives	o	Position(s)	pos
				Sequence(s)	seq
				Location(s)	loc
				Distance(s)	dist
				Machine(s)	M
				Line(s)	L
				Workstation(s)	WS
				Warehouse	WH
				Supermarket(s)	SUP
				Machine of storage/retrieve	S/R
				Equipment (s)	eq
				Worker(s)	wrk
				Operation (s)	op
				Job(s)	j
				Product(s)	pr
				nth product	npr
				Components	comp
				Capacity	cap
				Volume(s)	vol
				Quantity(ies)	qty
				Cycle(s)	cyc
				Tour (s)	tr
				Processing time	PT
				Takt time	TT
				Cycle time	CT
				Lead time	LT
				Delivery time	DT
				Waiting time	WT
				Replenishment time	RT
				Function	fct
				Constant	const
				Minimum	min
				Maximum	max
				Average	avg
				Probability	prob
				Proportion	prop
				Between	btw
				Within	$\in$
				Parallel	//
				Last-in-first-out	LIFO
				Maximum feasible elapsed time	MFET
				Bulk warehousing system	BWS
				Other warehousing system	OWS
				Mean square deviation	MSD
				Return on investment	ROI
				Net present value	NPV
				%Demand	dem
				Destination(s)	dest
				Begining	beg
				Rectilinear	rect
				Consecutive	csv
				Arrangement(s)	arr
				Enterprise resource planning	ERP
				Computerized maintenance management system	CMMS

Table B.2: Process features.

Features	Code	Features	Code	Features	Code	Features	Code
Level of automation	P.g.1	Lot size	P.p.18	Retrieval patterns (depletion rate)	P.w.19	Vertical interaction	P.a.18
Model type/L	P.g.2	Buffer Size	P.p.19	Info on the storage area and loc	P.w.20	the triangle inequality holds	P.a.19
Strategy of assembly	P.g.3	Subcontracting vol for npr/u (t)	P.p.20	Dist btw to pairs of pallets	P.w.21	Fixed or changeable storage size	P.a.20
Workflow characteristics	P.g.4	Allocation of pr to L (t)	P.p.21	Frequency of move	P.w.22	Only one aisle/ zone	P.a.21
New process technology	P.g.5	# of pr to be produced	P.p.22	Avg travel dist	P.w.23	Aisles are identical	P.a.22
Efficiency flexibility trade-off	P.g.6	# of pr produced in overtime/L	P.p.23	Material qty/pr from supply region (t)	P.δ.1	Cross aisle connect aisles end	P.a.23
Set of op	P.Π.1	Production plan/shift	P.p.24	# of pr from plant to sales region (t)	P.δ.2	Single, dual or multi-command	P.a.24
Set of picking tasks	P.Π.2	Allocation of available M cap to op	P.p.26	Transportation schedules	P.δ.3	Random deterioration	P.a.25
Set of waves	P.Π.3	Workload for each maintenance work (t)	P.p.27	Constraint of processing op/M	P.c.1	U-flow configuration	P.a.26
Set of orders	P.Π.4	Work orders	P.p.28	Max # of tr	P.c.2	Depleted based on LIFO	P.a.27
Set of loc of each order	P.Π.5	History file of interventions	P.p.29	Min order qty	P.c.3	Multiple pickers/zone at the same time	P.a.28
Facility layout config	P.β.1	Prediction of the maintenance load	P.p.30	Constraints of inclusion and exlusion	P.c.4	Batching or not	P.a.29
Process Planning	P.β.2	Overtime workload	P.p.31	Constraint of cunsumption of energy	P.c.5	Static or dynamic batching	P.a.30
The starting setup config on M	P.β.3	Subcontract workload	P.p.32	Precedence constraints btw tasks	P.c.6	The dist btw each storage pos are known	P.a.31
Maintenance organization	P.β.4	Backlog level	P.p.33	Weight of handling unit	P.c.7	Each storage pos is visited only once	P.a.32
Loading planning	P.β.5	Inspection schedule	P.p.34	Max # of op/M/shift	P.c.8	Stationary material flow over the PH	P.a.33
Picking pos/aisle	P.β.6	Upgrading maintenance j specifications	P.p.35	Picking cap/route in # of items	P.c.9	Level of partitionning	P.a.34
Vertical storage system	P.β.7	Changing maintenance schedules	P.p.36	The last tr n must end within the PH	P.c.10	progressive zoning or // zoning	P.a.35
# of floors	P.β.8	Total length of the route	P.s.1	Precedence constraints btw orders	P.c.11	The orders are picked one at a time	P.a.36
# of aisles	P.β.9	Optimal driver routes	P.s.2	Floor weight limits	P.c.12	Single I/O point for the picker in the WH	P.a.37
# of Lanes for each depth	P.β.10	# of tr	P.s.3	Max # of lanes for storage	P.c.13	The seq of orders is fixed or free	P.a.38
I/O config	P.β.11	Kanban, L Stocking , Sequencing, Kitting	P.s.4	Transportation qty	P.c.14	Euclidian or manahttan dist	P.a.39
Zone config	P.β.12	Periodic , continuous review policy	P.s.5	Total # of WS cyc/shift	P.c.15	Congestion or not	P.a.40
# of regular maintenance op	P.β.13	Parts each SUP should handle	P.s.6	Levels of system risk/reliability	P.c.16	Orders are known beforehand or not	P.a.41
# of orders/d	P.θ.1	# of parts/ comp box	P.s.7	Failure level	P.c.17	It is possible to divide a lot	P.a.42
Reorder points and order qty	P.θ.2	Preparation batch size	P.s.8	Work balance constraints	P.c.18	Lots are available at the beg of the shift	P.a.43
Qty/order/ item	P.θ.3	# and types of comp/vehicle trip	P.s.9	Limits of the backlog of maintenance work	P.c.19	Routing and wave length are used	P.a.44
Order arrival rate	P.θ.4	Total demand btw WS	P.s.10	Seq of maintenance j	P.c.20	Monitoring through perfect inspections	P.a.45
Backorder level for npr/u (t)	P.θ.5	Input and output flows	P.w.1	Requirement of all maintenance j	P.c.21	Next inspection depends on current one	P.a.46
MTS, MTO, ATO, ETO	P.p.1	Shipping and receiving schedules	P.w.2	Possibility of setup carryover in csv (t)	P.a.1	Stationary law for maintained system state	P.a.47
CODP	P.p.2	# of picked items/order/pallet	P.w.3	Lot sizing & scheduling (simultaneously)	P.a.2	Deterministic maintenance j standards	P.a.48
Total flow btw facilities (t)	P.p.3	Pick list size	P.w.4	Static or dynamic facility layout	P.a.3	Maintenance load fluctuates over time	P.a.49
Total flow btw M (t)	P.p.4	Pick wave schedule	P.w.5	Static or dynamic L layout	P.a.4	Quality of work is based on standards	P.a.50
Rect dist btw facilities	P.p.5	Aisle width	P.w.6	Re-arrangements of M in csv (t)	P.a.5	Work order is the source of data collection	P.a.51
Rect dist btw M centroids	P.p.6	Aisle orientation	P.w.7	Re-arrangements of facilities in csv (t)	P.a.6	Timely availability of feedback from CMMS	P.a.52
# of thresholds	P.p.7	Aisles depth	P.w.8	A task can be assigned to only one WS	P.a.7	Maximize the efficiency/L	P.o.1
# of tasks/ pr model	P.p.8	Storage lane depth	P.w.9	Task assignment restriction to M	P.a.8	Minimize the smoothness index	P.o.2
# of op/j	P.p.9	Stack height	P.w.10	Seq of tasks depends on WS config	P.a.9	Minimize the workflow interference	P.o.3
Production master schedule	P.p.10	Storage clearance btw pallets	P.w.11	WS served by one SUP are always csv	P.a.10	Minimize the prob of early loading	P.o.4
Assignement of tasks to WS	P.p.11	Storage sizes (t)	P.w.12	Production seq is known	P.a.11	Optimize the ordering intervals	P.o.5
Re-assignment of each task to WS	P.p.12	Dedicated, Random Storage	P.w.13	The same seq/production cyc	P.a.12	Minimize the avg # of trip/d	P.o.6
Dist btw loc	P.p.13	Single, Sequential, Concurrent picking	P.w.14	Each WS must lie on exactly one route	P.a.13	Maximize the storage efficiency	P.o.7
Work structure and control	P.p.14	S-shape, Return, Midpoint, Largest gap	P.w.15	Each route is assigned to one vehicle	P.a.14	Maximize the storage cap	P.o.8
Seq of op assigned on same M	P.p.15	Size of each batch	P.w.16	The rate of production is (in)finite	P.a.15	Minimize the total travel distance	P.o.9
Production seq	P.p.16	Assignment of orders to baches	P.w.17	Several pr can be manufactured (t)	P.a.16	Maximize throughput	P.o.10
Production vol	P.p.17	Seq and route of loc for picking items	P.w.18	Permutation flow shop system	P.a.17	Maximize plant availability	P.o.11

Table B.3: Product features.

Features	Code	Features	Code	Features	Code
# of different pr	Z.g.1	Matrix of possible allocation pr to L	Z.β.3	SKU of parts is the same in the system	Z.a.3
# of models	Z.g.2	Items loc for storage	Z.β.4	Shortage is not permitted	Z.a.4
# of j	Z.g.3	Assignment of comp to modes	Z.β.5	Variety of pr	Z.a.5
Material class and p types	Z.g.4	Required spares and material to meet the load	Z.β.6	No job spliting	Z.a.6
# of SKUs	Z.g.5	Inventory level (t)	Z.α.1	No job interuption	Z.a.7
# of items stored	Z.g.6	Stock availability	Z.α.2	Fixed # of pos/j	Z.a.8
Frequency of change of pr	Z.g.7	Inventory saving	Z.α.3	Comp are produced internally or from supplier	Z.a.9
Fragility of handled pr	Z.g.8	Inventory of pr in the beg of PH	Z.α.4	All-units discounts or incremental discounts	Z.a.10
Shelf life of pr or comp	Z.g.9	Inventory of pr at the end of (t)	Z.α.5	Each Comp is used in only one WS	Z.a.11
Info on the stored items	Z.g.10	Inventory for each comp	Z.α.6	Package type units or Unit loads	Z.a.12
Demand in # of bins at each WS/shift	Z.g.11	Safety stock for each comp and pr	Z.α.7	Dimension, shape and form of handling u	Z.a.13
Demand in bins u of each WS during TT	Z.g.12	Safety stock level at line-side	Z.α.8	No safety Stock	Z.a.14
Expected daily avg demand/comp	Z.g.13	Current line-side inventory	Z.α.9	Only pr info is known about stored items	Z.a.15
Comp demand of WS/L	Z.g.14	Item qty/delivery from supplier	Z.α.10	The storing loc of an item is given	Z.a.16
Qty discounts	Z.g.15	Characteristics of pr	Z.c.1	The picking loc of an item is given	Z.a.17
Prob that pr $\in$ order	Z.g.16	Max qty of pr produced/L	Z.c.2	Spare parts are available	Z.a.18
Set of pr	Z.Π.1	SKUs stackability limits	Z.c.3	Minimize the # of tardy j	Z.o.1
Set of pr types	Z.Π.2	Max inventory level	Z.c.4	Maximize qty of pr after nth setup on M (t)	Z.o.2
Set of models	Z.Π.3	Inventory constraints	Z.c.5	Minimize inventory level at each WS	Z.o.3
Set of comp	Z.Π.4	Storage assignment	Z.c.6	Minimize line-side stocks	Z.o.4
Set of all items	Z.Π.5	Weight of item u/pallet	Z.c.7	Minimize inventory error btw real and ERP	Z.o.5
Set of SKUs	Z.Π.6	Pr quality	Z.c.8	Maximize pr quality	Z.o.6
Prop of # of pr with specific type	Z.β.1	Single or multi-pr	Z.a.1		
% of end pr that use specific comp	Z.β.2	Const speed of pr move btw WS	Z.a.2		

Table B.4: Resource features.

Features	Code	Features	Code	Features	Code	Features	Code
# of facilities	R.g.1	Assignment of pickers to Sup	R.s.6	Bin weight filled with comp/tr	R.c.21	(De)centralized WH	R.a.27
# of M	R.g.2	AS/RS	R.w.1	Pallet cap	R.c.22	Simple or General Assembly L	R.a.28
# of WS/L	R.g.3	BWS,OWS	R.w.2	WH size	R.c.23	Only one L feeded by many SUP	R.a.29
# of WS	R.g.4	Dimension of storage racks	R.w.3	Storage layout	R.c.24	Only one central SUP or WH	R.a.30
# of L	R.g.5	Pallet placement angle w.r.t aisle	R.w.4	Cap of each zone	R.c.25	WH has enough parts to supply dem	R.a.31
# of Sup	R.g.6	# of racks/aisle	R.w.5	Zoning constraints	R.c.26	SUP has enough parts to supply dem	R.a.32
# of cranes	R.g.7	Height of a pallet of a SKU	R.w.6	Max labor level (t)	R.c.27	Central P&D point	R.a.33
# of zones	R.g.8	Pallet rack	R.w.7	Max WH space (t)	R.c.28	Bins are available at the beg PH	R.a.34
# of S/R	R.g.9	Size of pick bins	R.w.8	Carrier characteristics	R.c.29	Identical bins size	R.a.35
Set of M	R.Π.1	Containers size in an AS/RS	R.w.9	Max # of setups of M/shift	R.c.30	# and dest of bins are known	R.a.36
Set of eq	R.Π.2	WH config	R.w.10	Max # of M/op	R.c.31	Bin take the same place in stage	R.a.37
Set of zones	R.Π.3	Space consumed/u/pr	R.w.11	Zones ∈ shopfloor boundaries	R.c.32	// picking aisles	R.a.38
Set of pickers	R.Π.4	Space to lease (t)	R.w.12	Zones ∈ L boundaries	R.c.33	Only one container size is used	R.a.39
Set of pallets	R.Π.5	WH spaces of npr/u (t)	R.w.13	Each facility is assigned to one loc	R.c.34	Crane could serve multiple aisles	R.a.40
Set of bins	R.Π.6	Avg speed of vehicle	R.w.14	Each M is assigned to one loc	R.c.35	Only one op by S/R each time	R.a.41
Set of stages/cart	R.Π.7	MHE type	R.w.15	Total storage space	R.c.36	All zones are of the same size	R.a.42
Set of carriers	R.Π.8	Transport device	R.w.16	Each zone has at least one facility	R.c.37	Non overlapping zones	R.a.43
Set of docks	R.Π.9	# of Vehicles	R.w.17	Each zone has at least one M	R.c.38	Dependence of comp with other comp failures	R.a.44
Set of skills of wrk	R.Π.10	Assignment of tasks to pickers	R.w.18	Energy consumption of eq	R.c.39	Available manpower is known at the beg of PH	R.a.45
Set of maintained eq	R.Π.11	Picker's travel speed in zones	R.w.19	Manpower balance constraints	R.c.40	Some zones must have fixed pos	R.a.46
Allocation of eq to M	R.p.1	Length and width of zones	R.w.20	Eq condition	R.c.41	Available eq is known at the beg of PH	R.a.47
Space requirement of eq	R.p.2	Assignment of zones to SKUs	R.w.21	Multi ressource	R.a.1	Eq is 100% reliable	R.a.48
# of all loc	R.p.3	Assignment of pickers to zones	R.w.22	Single or multi-departments	R.a.2	Similar pickers skills	R.a.49
Eq tear	R.p.4	Loc of S/R	R.w.23	Sufficient cap for dem	R.a.3	Every loc has the same size	R.a.50
# of M loc	R.p.5	# of P&D WS	R.w.24	Seq dep tool change	R.a.4	SKU locs are given	R.a.51
Facility loc (t)	R.p.6	Dock loc	R.w.25	Level of ergonomy	R.a.5	Finite # of pickers	R.a.52
Machine loc (t)	R.p.7	Loc of the P&D	R.w.26	Sufficient space for facilities	R.a.6	Infinite walking speed	R.a.53
Flexibility reserve/L	R.p.8	Cap of L/pr (t)	R.c.1	Sufficient space for M	R.a.7	Availability of M/shift is given	R.a.54
Availability of 2nd L for pr (t)	R.p.9	# of cap expansion steps of L (t)	R.c.2	Const gap btw facilities	R.a.8	U load WH or U load AS/RS	R.a.55
Throughput/L	R.p.10	Max cap of L/pr	R.c.3	Const gap btw M	R.a.9	Only one j/M (t)	R.a.56
Hired wrk (t)	R.p.11	Prop of normal cap/L	R.c.4	Non-identical M	R.a.10	Tow train cap limited only by # of bins	R.a.57
Laid off wrk (t)	R.p.12	Cap available for M (t)	R.c.5	M arranged in // rows	R.a.11	Maximize the cap utilization	R.o.1
Loc of each M in next period	R.p.13	Cap of Sup	R.c.6	Const cap of plants constructed	R.a.12	Minimize the # of M	R.o.2
Avg deterioration speed	R.p.14	Vehicle load cap	R.c.7	Only one pr/L	R.a.13	Minimize the risk of fatigue	R.o.3
Productivity of skills	R.p.15	Pick cap of picker	R.c.8	Const throughput/L	R.a.14	Minimize the workload balance	R.o.4
Reliability of eq and tools	R.p.16	Cap of each stage	R.c.9	Only one op/M (t)	R.a.15	Minimize the # of vehicles	R.o.5
Maintenance strategy for each eq	R.p.17	Prob to not exceed the Sup cap	R.c.10	WS can perform any op	R.a.16	Maximize the vehicle utilization	R.o.6
# and skills of craftsmen	R.p.18	Min load in each L (t)	R.c.11	M setup completed (t)	R.a.17	Minimize the # of vehicle trips	R.o.7
Required maintenance tools	R.p.19	Max # of L/pr (t)	R.c.12	At any time, at most one j/M	R.a.18	Minimize the energy consumption	R.o.8
Resource planning for maintenance j	R.p.20	Zone boundaries of facility	R.c.13	M are set up for a predefined pr	R.a.19	Maximize the eq utilization	R.o.9
Training craftsmen	R.p.21	Zone boundaries of M	R.c.14	Rect moves of M for the relayout	R.a.20	Maximize the space utilization	R.o.10
Revising maintenance strategy of eq	R.p.22	Capable M/op	R.c.15	Vehicles perform cyc without break	R.a.21	Maximize the crane throughput	R.o.11
# of bins/comp/L	R.s.1	Space constraint of L	R.c.16	Identical cap for all vehicles	R.a.22	Minimize the picker effort	R.o.12
# of bins/WS/TT	R.s.2	# of pickers/zone	R.c.17	Const speed of vehicle	R.a.23	Maximize the avg # of bins/trip	R.o.13
Sup Loc	R.s.3	Shelf cap: max # of bins	R.c.18	Storage space indep of modes	R.a.24	Minimize the # of zones visited	R.o.14
# of drivers/delivery	R.s.4	Total storage cap	R.c.19	Storage eq	R.a.25	Maximize eq availability	R.o.15
Assignment of driver to WS	R.s.5	SKU capacity	R.c.20	Shape of WH	R.a.26		

Table B.5: External effect features.

Features	**Code**
Demand for pr in sales region (t)	E.μ.1
Demand requirements of pr	E.μ.2
Distribution of the demand (t)	E.μ.3
Forecasted demand for npr/u (t)	E.μ.4
Set of sales regions	E.Π.1
Competitive pressure	E.c.1
Competitors strategy	E.c.2
Environmental and safety restrictions	E.c.3
Government regulation	E.c.4
Sevrice requirements	E.c.5
Service level	E.c.6
Safety laws	E.c.7
Safety limit	E.c.8
Single site-companies	E.a.1
Demand patterns for pr are indep	E.a.2
Specific distribution of daily historical demand	E.a.3
Known or Uncertain demand	E.a.4
Deterministic or stochastic demand	E.a.5
Skewness of demand	E.a.6
Full or partial backordering of excess demand	E.a.7
Backordered demand must be fulfilled in the next (t)	E.a.8
Fixation or fluctuation in price	E.a.9
Maximize market share	E.o.1
Maximize service level	E.o.2

Table B.6: Cost features.

Features	Code	Features	Code	Features	Code
Control costs	C.g.1	Pr cost	C.p.15	Dist-dep tow train travel costs	C.a.7
System costs	C.g.2	Pr selling price	C.p.16	Const operator cost rate/h	C.a.8
Acquisition cost	C.g.3	Machine shifting cost btw loc (t)	C.p.17	Escalating factors are certain over PH	C.a.9
Annual discount rate	C.g.4	M installation cost (t)	C.p.18	Minimize investment costs	C.o.1
Fixed expenditure	C.γ.1	Cost/inspection	C.p.19	Minimize facilities installation costs	C.o.2
Cost of investment	C.γ.2	Cost/corrective replacement	C.p.20	Minimize MHC	C.o.3
Back order cost	C.θ.1	Cost/preventive replacement	C.p.21	Minimise the rebalancing costs	C.o.4
Order costs	C.θ.2	Long run maintenance cost/ ut	C.p.22	Minimize cost of rearrangement	C.o.5
Inventory cost of u/WS/cyc	C.α.1	Operating cost/ut	C.p.23	Minimize the total cost	C.o.6
Inventory holding cost	C.α.2	Cost of one man-h of skill	C.p.24	Minimize NPV of cash flows/y	C.o.7
Inventory replenishment cost	C.α.3	Cost of backloging one man-h (t)	C.p.25	Minimize transportation and inventory costs	C.o.8
Penalty costs	C.α.4	Cost and quality reports	C.p.26	Minimize the total weighted dist costs	C.o.9
Production cost of pr/L (t)	C.p.1	Moving cost of u/bin/dist	C.s.1	Minimize the fixed costs of the SUP	C.o.10
Production cost of pr/M (t)	C.p.2	Cost of vehicle/shift	C.s.2	Minimize costs/ut	C.o.11
Production cost of npr/u (t)	C.p.3	Comp cost	C.w.1	Minimize the total inventory holding cost	C.o.12
Overtime production cost/pr/L	C.p.4	WH construction cost	C.w.2	Maximize the ROI	C.o.13
Costs to hire wrk (t)	C.p.5	MHE cost	C.w.3	Minimize the acquisition cost	C.o.14
Costs to layoff wrk (t)	C.p.6	Material supplying cost	C.δ.1	Minimize the operating cost	C.o.15
Subcontracting cost of npr/u (t)	C.p.7	Distribution cost of pr (t)	C.δ.2	Minimize inventory and storage costs	C.o.16
Escalating factors of cost categories	C.p.8	Available budget	C.c.1	Minimize the construction cost	C.o.17
Eq cost	C.p.9	Cost is dep or indep from WS and tool	C.a.1	Minimize the maintenance cost	C.o.18
Tool change cost	C.p.10	Const building cost	C.a.2	Minimize the traveling costs	C.o.19
Facility construction cost	C.p.11	Redesign cost is fct of rectilinear dist of moves	C.a.3	Minimize the long run expected maintenance cost/ut	C.o.20
Opening cost/cap	C.p.12	Fixed cost/supermarket	C.a.4	Minimize the shut-down cost	C.o.21
Facility installation cost (t)	C.p.13	Material flow costs=dist*cost/u.dist	C.a.5		
Holding cost of pr	C.p.14	Seq dep setup costs	C.a.6		

Table B.7: Time features

Features	Code	Features	Code	Features	Code
Set of periods	T.g.1	WH throughput in ut	T.w.1	Stochastic loading & unloading times	T.a.15
# of periods	T.g.2	Inventory CT	T.w.2	Seq dep setup times	T.a.16
PH	T.g.3	# of pallets/ut	T.w.3	Setup times are negligible	T.a.17
Daily working time	T.g.4	Length of a storage aisle in ut	T.w.4	The stopover time is const and known	T.a.18
Time profile of pr investment	T.γ.1	Travel time btw storage aisles	T.w.5	The RT is const and known	T.a.19
Time profile of structural investment	T.γ.2	Set-up time of a pick route	T.w.6	The driving time is const and known	T.a.20
PT of j/M	T.p.1	Item picking time	T.w.7	Stopover time is indep of the # of bins	T.a.21
Task time	T.p.2	Additional time for other op	T.w.8	Fixed order LT	T.a.22
Planned production LT	T.p.3	Transportation time	T.w.9	Const loading & unloading times	T.a.23
LT	T.p.4	Batch time window	T.w.10	Depletion times are null	T.a.24
Maintenance j arrival times	T.p.5	Times related to AS/RS movements	T.w.11	Travel time btw zones is negligible	T.a.25
CT/pr	T.p.6	Required delivery LT	T.c.1	Standard pick time for all zones	T.a.26
DT of spare parts	T.p.7	TT/L	T.c.2	Arrival time of items is known	T.a.27
Total idle time/WS/L	T.p.8	Min vehicle CT	T.c.3	Departure time of items is known	T.a.28
Priority and date work	T.p.9	Downtime limit	T.c.4	Distribution of arrival time of orders	T.a.29
Starting and completion time of pr/M (t)	T.p.10	Min time fixed with the supplier	T.c.5	Parameter values are certain over PH	T.a.30
Regular time production of npr (t)	T.p.11	Time btw two visits not exceed MFET	T.c.6	Only discrete set off times of vehicles	T.a.31
Overtime production of npr (t)	T.p.12	Required order DT	T.c.7	Negligible time of loading/unloading	T.a.32
M availability time/shift	T.p.13	Picker CT	T.c.8	Negligible time for the replacement	T.a.33
Completion time	T.p.14	Time slots	T.c.9	Random maintenance j arrival times	T.a.34
Amount of WIP in mn of op/M	T.p.15	Order due dates	T.c.10	Minimize TT	T.o.1
Tool loading time	T.p.16	Setup-time of M	T.c.11	Minimise CT	T.o.2
Tool unloading time	T.p.17	Man-h availability limits	T.c.12	Minimize makespan	T.o.3
Matrix of PT of eq/tasks	T.p.18	Overtime man-h limit	T.c.13	Minimize the flow time	T.o.4
Hours of labor of npr/u (t)	T.p.19	# of overtime-subcontracted wrk (t)	T.c.14	Minimize the max tardiness	T.o.5
Hours of machine of npr/u (t)	T.p.20	Max % of overtime/regular h	T.c.15	Minimize MSD of completion times	T.o.6
Mean-time btw failures	T.p.21	Discrete time	T.a.1	Minimize the idle time of pr/M (t)	T.o.7
Mean-time to repair	T.p.22	(In)finite PH	T.a.2	Minimize # of delays	T.o.8
Time to complete a maintenance j	T.p.23	Random standard times	T.a.3	Minimizes total route duration	T.o.9
# of h made available for maintenance cap	T.p.24	Multi-period	T.a.4	Minimize avg RT	T.o.10
Avg # of bins/ut	T.s.1	Time is (in)dep from WS and tool	T.a.5	Minimize avg supply LT	T.o.11
RT	T.s.2	DT for spare parts is negligible	T.a.6	Minimize WT for drivers	T.o.12
SKU loading time	T.s.3	Deterministic or stochastic task times	T.a.7	Minimize CT of stockage and retrieving	T.o.13
SKU unloading time	T.s.4	Discrete time of possible interventions	T.a.8	Minimize the system throughput time	T.o.14
Vehicle start time (/tr)	T.s.5	Task times are known	T.a.9	Minimize picking time	T.o.15
Vehicle CT	T.s.6	PT and moving time are ignored	T.a.10	Minimize travel time	T.o.16
Travel time btw WS	T.s.7	PT/lot size at each op is known/shift	T.a.11	Minimize the idle time of craftsmen	T.o.17
Service time in each WS	T.s.8	Identical PT on all M	T.a.12	Minimize plant shut-down time	T.o.18
MFET btw two visits to the WS	T.s.9	Travel times of operators are ignored	T.a.13	Minimize total scheduling time	T.o.19
Replenishment LT	T.s.10	Deterministic loading & unloading times	T.a.14		

Table B.8 uses the notation "*Class.Category.NumberID*" to specify the CIAOC properties of each problem decision and provide key references. The first line corresponds with the decision problem ID=1, i.e., by looking at the X, Production Strategy. Among its features, we found as constraints E.c.1 and E.c.6, then, C.g.2, and finally T.p.3 and T.c.1. According to the Table B.5, the first two are *Competitive pressure* and *Service level*, Table B.6 gives the third one *System costs*, while Table B.7 provides details of the last two constraints, which are *Processing time of job/Machine* and *Planned production Lead-Time* and *Required delivery Lead-Time*. These features are reported by Atan et al. (2017); Ghalehkhondabi et al. (2016); Olhager (2010).

[H]

Table B.8: Properties of decision problems.

ID	Constraint	Input	Assumption	Output	Criteria	References
1	E.c.1; E.c.6; C.g.2; T.p.3; T.c.1	Z.Π.1; E.μ.2; C.p.15	E.a.1; E.a.4	P.p.1-2	P.g.6; Z.α.2-3; E.o.2; C.θ.1; C.p.1; C.p.14; C.p.16	(Atan et al., 2017; Ghalehkhondabi et al., 2016; Olhager, 2010)
2	P.p.2; Z.g.1; Z.β.3; Z.c.2; R.c.1-3; R.c.11-12; C.γ.1; C.c.1; E.c.3-4	P.g.5; Z.g.4; Z.Π.1; Z.β.1; R.p.8; R.c.4; E.μ.1; E.μ.3; E.Π.1; E.c.2; C.g.4; C.p.1; C.p.4; C.p.11-12; C.δ.1-2; T.γ.1-2	P.p.1; R.a.1; R.a.12; E.a.1; T.a.1-2; T.a.4	P.p.21-23; P.δ.1-2; R.p.9; R.c.2	R.o.1; E.o.1; C.o.6-7	(Ceryan and Koren, 2009; Martínez-Costa et al., 2014; Niroomand et al., 2012)
3	Z.c.5; R.c.5; R.c.27-28	P.θ.5; P.p.20; Z.Π.1; Z.α.1; R.w.13; E.μ.4; C.θ.1; C.α.2; C.p.3-8; T.p.11-12; T.p.19-20	Z.a.1; E.a.10; C.a.9; T.a.2; T.a.30	P.p.17; R.p.11-12	C.o.6	(Chakrabortty and Hasin, 2013)
4	P.c.8; R.c.30-31; T.p.13	P.Π.1; P.p.24; R.Π.1	P.a.42-43; R.a.10; R.a.54; T.a.11-12	P.p.26; T.p.15	P.o.10	(Toktay and Uzsoy, 1998)
5	P.g.4; R.c.16; C.c.1; T.p.2	P.g.1; P.β.2; P.p.17; Z.a.5	R.a.28	P.g.2-3	C.o.13	(Thomopoulos, 2014; Ullah et al., 2014)
6	P.c.4; P.c.6; R.g.2	P.g.2-3; Z.Π.1; Z.Π.3; T.p.2	P.a.7; R.a.28; T.a.7	P.p.11	P.o.1; R.o.2; T.o.1	(Battaïa and Dolgui, 2013; Boysen et al., 2007, 2008)
7	P.p.11; P.c.4-5; R.g.2; R.p.2; C.c.1; T.p.2	Z.Π.1; Z.Π.3; R.Π.2; C.p.9-10; T.p.16-18	P.a.9; R.a.4-5; C.a.1; T.a.5; T.a.7; T.a.14-15	R.p.1	R.o.3; R.o.8; C.o.1	(Özdemir and Ayağ, 2010; Wilhelm, 1999)
8	R.c.13; R.c.32; R.c.34; R.c.37	P.p.3; P.p.13; R.g.1; R.p.3; C.p.13; T.g.2	P.p.5; P.a.3; P.a.6; R.a.6; R.a.8; R.a.42-43; R.a.46; E.a.2; C.a.3; C.a.5	P.β.1; R.p.6	C.o.2-3	(De Koster et al., 2007; Kulturel-Konak, 2007; Saraswat et al., 2015)
9	R.c.14; R.c.33; R.c.35; R.c.38	P.p.4; P.p.13; R.g.2; R.p.5; C.p.18; T.g.2	P.p.6; P.a.4-5; R.a.7; R.a.9; R.a.11; R.a.42-43; E.a.2; C.a.5; T.a.10	R.p.7	P.o.3; C.o.3	(Battini et al., 2007; Ko et al., 2005)
10	P.p.11; P.c.4; P.c.6; R.g.2; T.c.2	P.p.8; P.p.18; Z.g.2; R.g.3; R.g.5; T.p.2; T.p.8	P.g.1-3; P.a.8; R.a.13; R.a.16; T.a.7; T.a.13	P.p.12	P.o.1-2; R.o.4; C.o.4; T.o.2	(Belassiria et al., 2018; Koren et al., 2018; Sancı and Azizoğlu, 2017)
11	R.p.7; R.c.14; R.c.33; R.c.35; R.c.38	R.p.5; R.g.2; T.g.2; P.p.4; P.p.13; C.p.17	P.p.6; P.a.4-5; R.a.7; R.a.9; R.a.11; R.a.20; R.a.42-43; E.a.2; C.a.5; T.a.10	R.p.13	C.o.5	(Bortolini et al., 2019; Kovács and Kot, 2017)
12	P.c.16; Z.c.8; R.c.39; E.c.8	Z.a.5; R.p.4; T.p.21-22	R.a.44	P.β.4; P.p.14; R.p.17	R.o.8-9; C.o.1; C.o.13; C.o.18	(Shafiee, 2015)
13	P.c.17	P.p.7; R.p.14; R.p.17; C.p.19-23	P.a.25; P.a.45-47; T.a.8; T.a.33	P.p.30	C.o.18; C.o.20	(Grall et al., 2002)
14	P.c.18-19; T.c.12-15	P.p.27; P.p.30; R.Π.10; R.p.11; R.p.15; C.p.5; C.p.24-25; T.g.3-4	T.a.3; T.a.34	P.p.31-33; Z.β.6; R.p.18-19; T.p.24	R.o.9; C.o.6; C.o.18; T.o.17	(Al-Shayea, 2012)
15	P.p.10; P.c.20-21; R.c.40	Z.α.1; R.p.16; R.p.18; T.p.5; T.p.7; T.p.9; T.p.23	P.a.48; Z.a.18; R.a.45; R.a.47-48; T.a.6	P.p.34; R.p.20	R.o.15; C.o.21; T.o.8; T.o.17-19	(Ben-Daya et al., 2016)

ID	Constraint	Input	Assumption	Output	Criteria	References
16	P.p.17; Z.c.8; R.c.41; T.p.24; T.c.4	P.p.28-29; R.Π.11; C.p.26	P.a.49-52	P.p.35-36; R.p.21-22	P.o.11; Z.o.6	(Gits, 1994)
17	P.p.15; P.p.19; P.c.1; P.c.6; R.c.15; T.p.14; T.c.11	P.p.9; Z.g.3; R.g.2; T.g.1-2; T.p.1	Z.a.6-8; R.a.18; T.a.9; T.a.16-17	P.p.16; T.p.10	Z.o.1; T.o.3-6	(Demir and İşleyen, 2013; Prabaharan et al., 2006; Udhayakumar and Kumanan, 2010)
18	R.c.5	P.β.3; Z.g.1; Z.Π.2; Z.α.4; R.g.2; R.Π.1; E.μ.4; C.p.2; C.p.14; T.g.1-2; T.p.1	P.a.1-2; P.a.16-19; Z.a.4; R.a.15; R.a.17-19; R.a.56; E.a.4-5; C.a.6; T.a.16	Z.α.5; T.p.10	Z.o.2; C.o.6; T.o.7	(Maravelias and Sung, 2009; Martínez et al., 2018; Ramezanian et al., 2013)
19	R.c.7; R.c.20; T.g.4	Z.g.2; Z.g.13; Z.Π.1; Z.Π.4; R.g.4-6; R.p.10; R.w.15; C.w.1	Z.a.3; R.a.14; R.a.23; R.a.27; E.a.3; T.a.17	R.w.17	Z.o.3; R.o.5-6; T.o.8	(Emde and Boysen, 2012b; Faccio et al., 2013)
20	R.c.6; R.c.10	P.s.10; Z.g.2; Z.g.11; R.g.4-5; R.p.3; R.p.7; R.p.10; C.s.1	P.a.10-11; R.a.27; R.a.29; R.a.35; C.a.4; C.a.7	P.s.6; R.g.6; R.s.3	C.o.8-10	(Alnahhal and Noche, 2015b; Battini et al., 2010; Emde and Boysen, 2012a)
21	P.p.16; R.c.7; R.c.18; T.c.6	Z.g.2; R.g.4; R.g.6; R.s.2; R.w.17; C.α.1; C.s.2; T.s.1-2; T.s.7-9	P.a.13-14; R.a.21-22; R.a.30-32; R.a.35	P.s.2; R.s.5	Z.o.4; T.o.9	(Emde and Boysen, 2012b; Emde and Schneider, 2018; Satoglu and Sahin, 2013)
22	P.c.2; P.c.10; P.c.15; R.c.7; R.c.18; T.c.3	Z.g.12; Z.Π.4; Z.α.9; R.w.17; T.g.3	R.a.31-32; R.a.57; E.a.4; T.a.18-21; T.a.31	P.s.3; T.s.5-6	P.o.4; R.o.7	(Emde and Gendreau, 2017; Fathi et al., 2015; Golz et al., 2012)
23	Z.g.14; R.s.3-4; R.c.6-7; T.c.2	P.s.6-8; Z.Π.4; Z.β.2; R.w.14; R.w.16	Z.a.2; Z.a.11; R.a.24; C.a.8	P.s.4; Z.β.5	C.o.15	(Battini et al., 2009; Sali and Sahin, 2016; Schmid and Limère, 2019)
24	Z.g.9; C.c.1; T.p.4	P.p.1; P.s.4; Z.Π.1; Z.Π.4; Z.α.4; Z.α.6; C.g.1; C.γ.2	Z.a.4; Z.a.9; R.a.27; E.a.4	P.s.5	P.o.5; Z.o.5	(Alnahhal and Noche, 2015a; Emde et al., 2012; Faccio et al., 2013)
25	P.p.1; P.c.3; Z.α.7; Z.c.4; R.c.19; T.c.5	P.β.5; P.p.24; P.s.5; Z.g.15; Z.Π.1; Z.Π.4; R.w.11; E.μ.3; C.θ.2; C.α.2; C.α.4; C.p.16; T.g.3; T.p.6; T.s.10	P.a.12; P.a.15; Z.a.4; Z.a.9-10; R.a.3; E.a.4; E.a.7; E.a.9; T.a.2; T.a.22	P.θ.2	E.o.2; C.o.11	(Rizkya et al., 2018; Singha et al., 2017; Toha et al., 2019)
26	Z.α.8; R.s.6; R.c.7; R.c.18; R.c.20-21; T.g.4	P.s.1; Z.g.2; Z.g.13; Z.Π.4; Z.α.9; R.g.4-6; R.Π.6-7; R.s.1; R.s.3; R.w.14; R.w.17; R.c.9; T.s.3-4	R.a.23; R.a.34; R.a.36-37; T.a.21; T.a.23	P.s.9	P.o.6; Z.o.4; R.o.13; C.o.12; T.o.10-12	(Nahmias and Olsen, 2020; Perera et al., 2017; Singha et al., 2017)
27	P.c.12; R.c.36; C.c.1; T.c.7	P.g.1; P.β.8; P.θ.1; P.θ.3; Z.g.6-7; Z.α.10; C.g.3; T.w.1	R.w.15	P.β.7; R.w.1-2; R.w.7; R.a.25	C.o.14-15	(Bouh and Riopel, 2016; Roodbergen and Vis, 2009)

ID	Constraint	Input	Assumption	Output	Criteria	References
28	P.c.7; R.c.29; E.c.7; C.c.1; T.c.2	P.g.1; P.g.4; P.β.13; P.θ.1; P.θ.3; P.w.2; P.w.22-23; Z. g.8; Z.α.10; C.w.3	Z.a.12-13; R.a.25	R.w.16	R.o.9; C.o.15	(Bouh and Riopel, 2016; Chakraborty and Banik, 2005; Mohsen, 2010)
29	P.β.1; R.p.6; C.c.1	Z.Π.1; E.μ.2; C.α.2-3; C.w.2; T.g.3	P.w.13; P.a.20; P.a.24; Z.a.1; R.a.2; E.a.5; T.a.2	P.w.12; R.w.12; R.c.19	P.o.9; C.o.16-17	(Cormier and Gunn, 1996; Gu et al., 2010; Heragu et al., 2005)
30	R.c.23; R.c.36; E.c.6	P.w.23; Z.g.16; Z.c.1; R.c.19	P.a.24; P.a.26; P.w.13; R.a.26; R.a.33; R.a.38; R.a.55	P.w.6-7; P.β.9; R.w.25-26; T.w.4	P.o.7-9; R.o.10; C.o.3	(Cardona et al., 2015; Gue and Meller, 2009; Roodbergen et al., 2014)
31	P.c.13; Z.c.3-4; R.c.36; E.c.8	P.w.6-7; P.w.19; Z.g.5; R.w.6; T.w.2-3	P.a.27; Z.a.14; T.a.24	P.β.10; P.w.9-11; R.w.4	P.o.8; R.o.10; C.o.3	(Derhami et al., 2016)
32	R.c.36; E.c.6	P.β.9; P.w.23; Z.Π.6; R.g.9; R.w.5; R.w.23-24; R.w.26; T.w.11	R.w.25; R.a.39-41; T.a.32	P.β.11; R.g.7; R.w.3; R.w.9	P.o.8; R.o.9; C.o.3; C.o.17-18; T.o.13	(Roodbergen and Vis, 2009; Roodbergen et al., 2014)
33	P.c.9; R.c.8; R.c.23-24	P.β.9; P.Π.4; Z.Π.5; R.Π.4; T.w.4-6; T.w.9	P.w.13-14; P.a.21-22; P.a.28; P.a.35; R.a.42; T.a.25	R.g.8; R.w.20; R.c.17; R.c.25	T.o.14	(De Koster et al., 2012; Le-Duc and De Koster, 2005; Van Gils et al., 2016)
34	R.c.17; R.c.23-25	P.Π.4; P.β.9; Z.Π.5-6; R.Π.3-4; T.g.3; T.w.4-6	P.w.13; P.w.15; P.a.28-29; P.a.35; R.a.49; T.a.26	R.w.21-22	R.o.4; R.o.11; R.o.14	(De Koster et al., 2012; Van Nieuwenhuyse and De Koster, 2009; Yu and De Koster, 2009)
35	R.c.19; R.c.24; E.c.5	P.s.5; P.w.1; E.a.6	P.w.14; P.a.24; R.a.55	P.w.13	C.o.19; T.o.16	(Ang et al., 2012; Gu et al., 2007)
36	P.β.12; Z.c.6; R.c.24	P.θ.1; Z.c.1; R.w.10	P.a.29; P.a.44	P.w.14	P.o.7	(Parikh and Meller, 2008; Petersen II, 2009)
37	R.c.19; R.c.23-24	P.β.6; P.β.9; P.w.4; P.w.8; E.a.8	P.w.13-14; P.a.31-32; P.a.39; R.a.50	P.w.15	P.o.9; C.o.15; T.o.15	(Cano et al., 2017; Masae et al., 2020; Petersen, 1997)
38	P.w.14; Z.c.1; Z.c.4; R.c.8; R.c.26; T.c.8	P.β.11; P.w.20; Z.g.10-11; Z.Π.5; R.c.23-24	P.a.33; Z.a.15; T.a.27-28	Z.β.4	T.o.16	(Gu et al., 2007; Rojas-Reyes et al., 2019; Yang et al., 2013)
39	P.c.11; R.c.22; R.c.24	P.Π.4; P.θ.4; P.w.3; P.w.13-15; P.w.21; Z.c.7; R.Π.5; R.w.26; R.a.25; T.w.7-8	P.a.22-23; P.a.36-40; Z.a.16-17; R.a.38; R.a.51	P.w.5; P.w.18	R.o.12; C.o.3; C.o.15; T.o.16	(Çelk and Süral, 2013; De Koster et al., 2007; Scholz et al., 2016)
40	P.c.14; R.c.8	P.Π.4-5; P.w.5; R.w.8; R.w.10; R.w.19	P.a.30; P.a.34; P.a.36	P.w.16-17; T.w.10	R.o.12; R.o.4; T.o.16	(Cergibozan and Tasan, 2016; Henn et al., 2012; Yu and De Koster, 2009)

ID	Constraint	Input	Assumption	Output	Criteria	References
41	P.c.14; R.c.8	P.Π.2-3; P.w.5; P.δ.3; R.Π.4; R.Π.8-9	P.w.14; P.a.41; R.a.52-53; T.a.29	R.w.18	C.o.15	(Hong et al., 2012, 2015; Rubrico et al., 2011)

Appendix C

Conceptual Models

This appendix defines the partial integrated models of the four views : Function, Decision, Information, and Performance. Also, it defines the conceptual models of the views: Process, Data, and Resource & Organization. Further, it defines the integrated model of all conceptual models of views.

C.1 Conceptual Models of Views: Metamodels

C.1.1 Conceptual Model of Partial Integration of Views of an MS

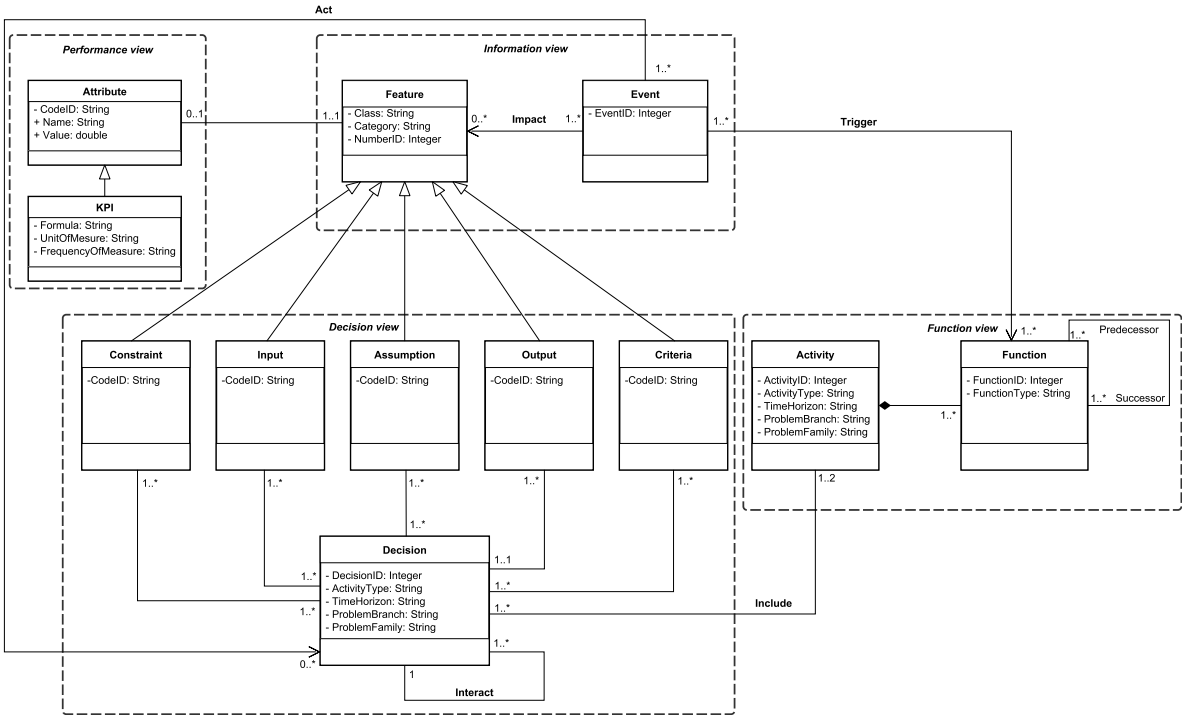

Figure C.1: Partial integration of views.

Figure C.1 shows that event triggers functions, impacts features, and acts on decisions. We explained the integration between performance and information views in Figure 3.7. Heritance relationship between CIAOC properties and feature means that each decision property is an MS feature. An association relationship between information and function views shows that occurred event triggers one or multiple functions to ensure the integration. Finally, an association relationship between event and decision explains that an event could act or not on many decisions.

C.1.2 Conceptual Model of Process View of an MS

Figure C.2 illustrates the process view conceptual model of an MS, consisting of multiple plants with various manufacturing systems and configurations. The work center serves as an abstraction of line and cell, allowing flexibility in including or excluding other work centers.

Abstract products generalize blocks and products, where blocks can be composed of other blocks, and products are combinations of blocks and/or products. Abstract products are associated with a product family and have a bill of materials containing components with specific quantities. Unique production properties, i.e., quantities and sequence orders for abstract products to launch in the work center, enable flexible planning.

Manufacturing orders are created for producing abstract products with specific quantities and temporal constraints. The operating sequences are an ordered list of operations that can be allocated to abstract products, but only one operating sequence is associated with the manufacturing order when created. Operations consume a specified quantity of components. Each component has one support, but support can be associated with one or many components. Preparation orders are generated for feeding components at the work center, and the picking list consolidates these orders.

Enumerations define possible values for types and statuses. The process view is interconnected with the function view since an MS encompasses diverse activities as outlined in Chapter 2.

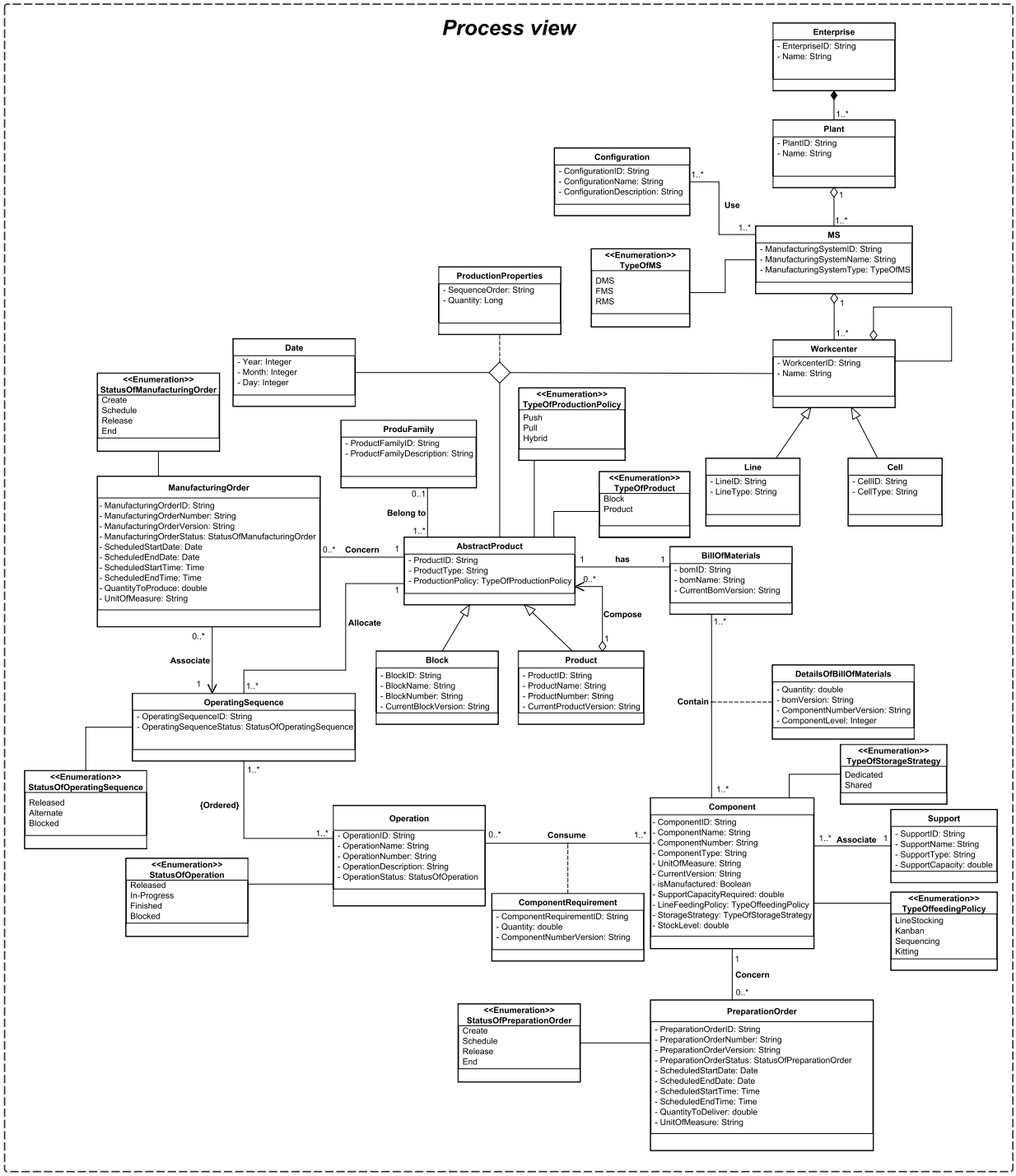

Figure C.2: Conceptual model of the process view of an MS.

C.1.3 Conceptual Model of Data View of an MS

Figure C.3 illustrates the conceptual model of the data view of an MS. The model shows that the data set collection is either provided from the enterprise systems (e.g., ERP, MES, WMS) or directly from messages perceived by smart assets. The message

respects a specific template and uses one Protocol that must be compatible with the protocols supported by the smart asset. Lastly, based on the data set collection, the data corresponding to the function is generated. We use a common object to have a standard structure of the function data. Furthermore, it could incorporate general data, simulation data, and any other data not specific to the function.

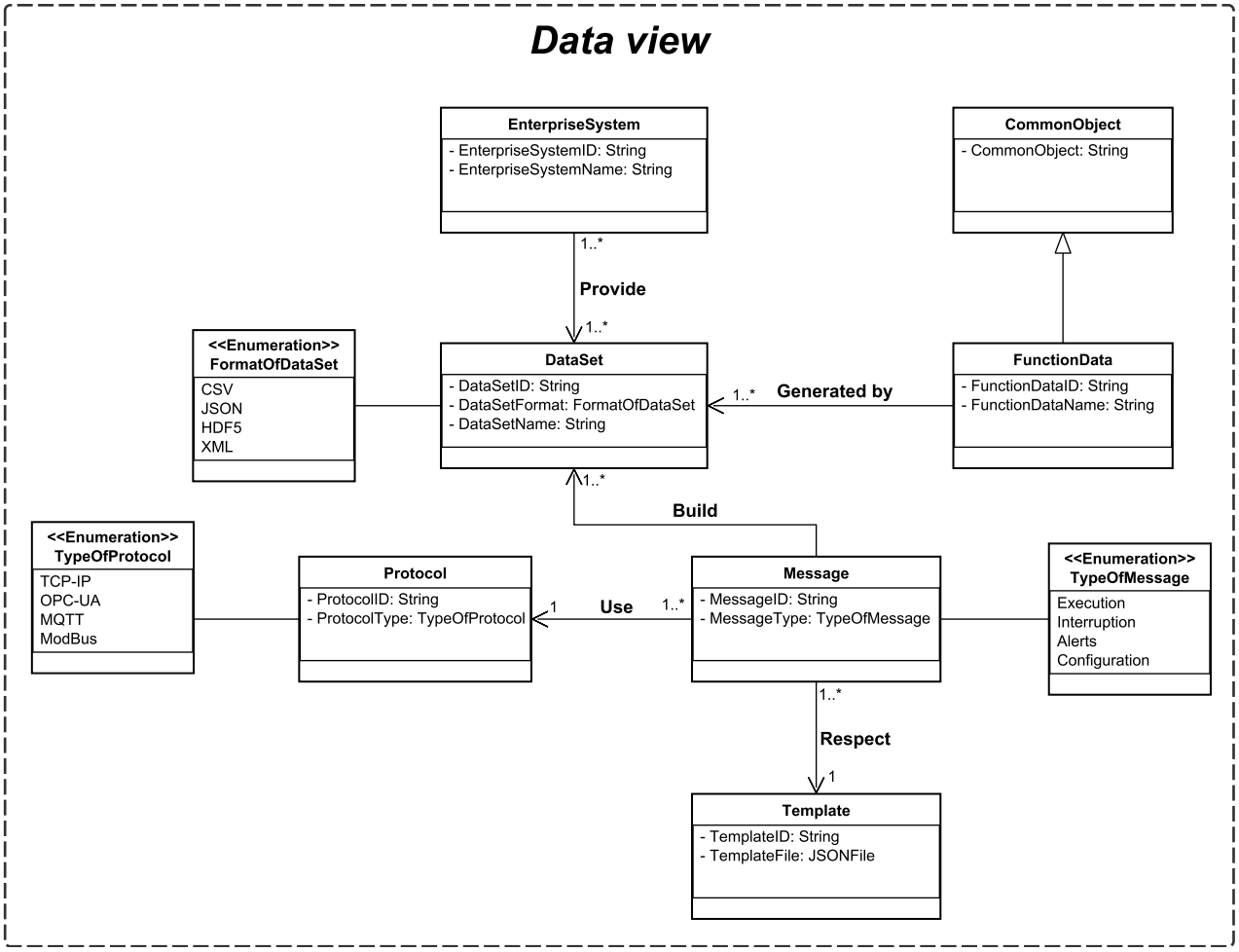

Figure C.3: Conceptual model of the data view of an MS.

C.1.4 Conceptual Model of Resource & Organization View of an MS

Figure C.4 illustrates the conceptual model of resource & organization view. Resources own many capabilities and are characterized by properties. Two types of resources are considered: spaces and production resources. Spaces are occupied by stations, which can be workstations, supermarkets, or picking stations. On the one hand, production resources can be equipped with smart assets like sensors, actuators, and exoskeletons. They encompass workers, machines, tools, and fixtures. Workers can be either operators or preparators, with preparators dedicated to one or multiple storage areas. On the other hand, production resources can work different shifts on specific dates and can be assigned to one or multiple workstations. Shifts consist of multiple calendar lines.

Carts are composed of stages, and picking stations are allocated to specific storage areas and can accommodate multiple carts.

The concept of a loop is used to represent an ordered list of stations visited by a transporter. Transporters are a generalization of AGVs, milk runs, and conveyors. They can perform multiple loops and tow multiple carts in each loop.

Typically, an enterprise is organized into several departments, which are composed of organizational units. Actors belong to a department (hierarchical dependence) and work in an organizational unit (functional dependence). Note that the organizational unit of an actor may not be within its department. Actors may supervise others, possess specific skills, and perform multiple roles, where each role may require multiple skills.

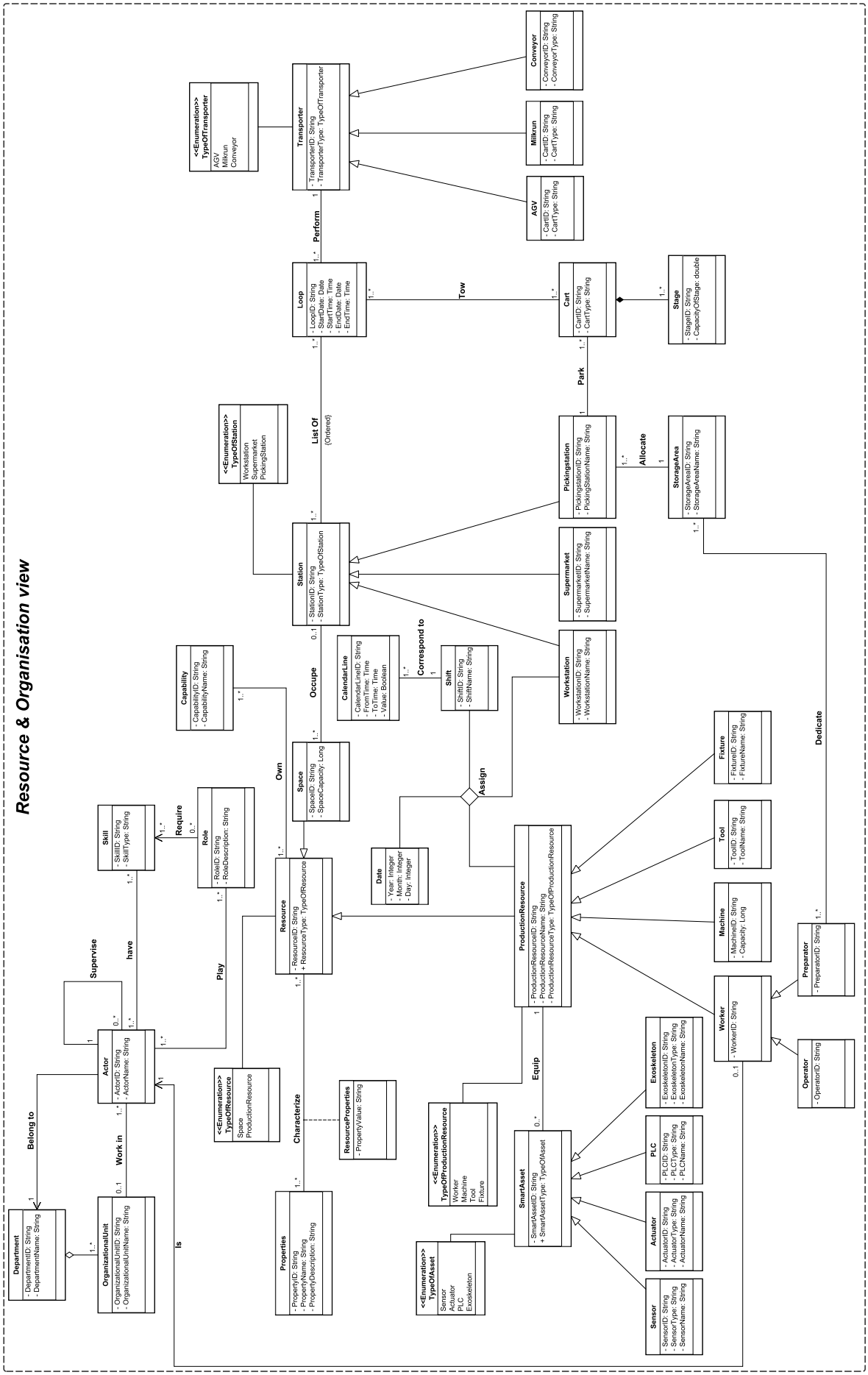

Figure C.4: Conceptual model of resource & organization view of an MS.

C.1.5 Conceptual Model of Integration of All Views of an MS

So far, we have defined all conceptual models of different views. We propose integrating the process, data, and resource & organization views with the partially integrated model (see Figure C.1). Figure C.5 illustrates the conceptual integrated model.

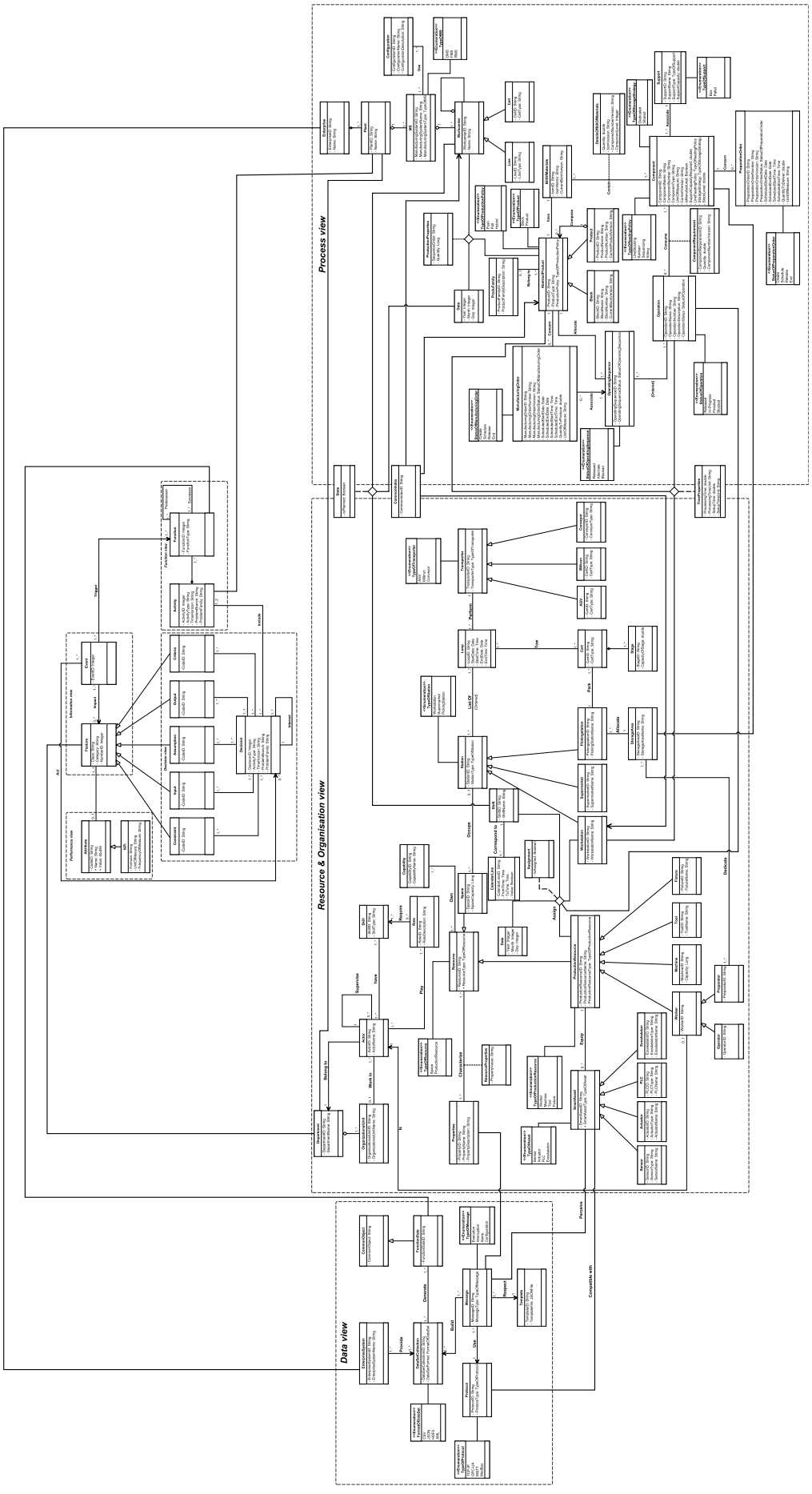

Figure C.5: Integrated conceptual model of an MS.

C.2 Conceptual Models of MEMO I4.0 Principles

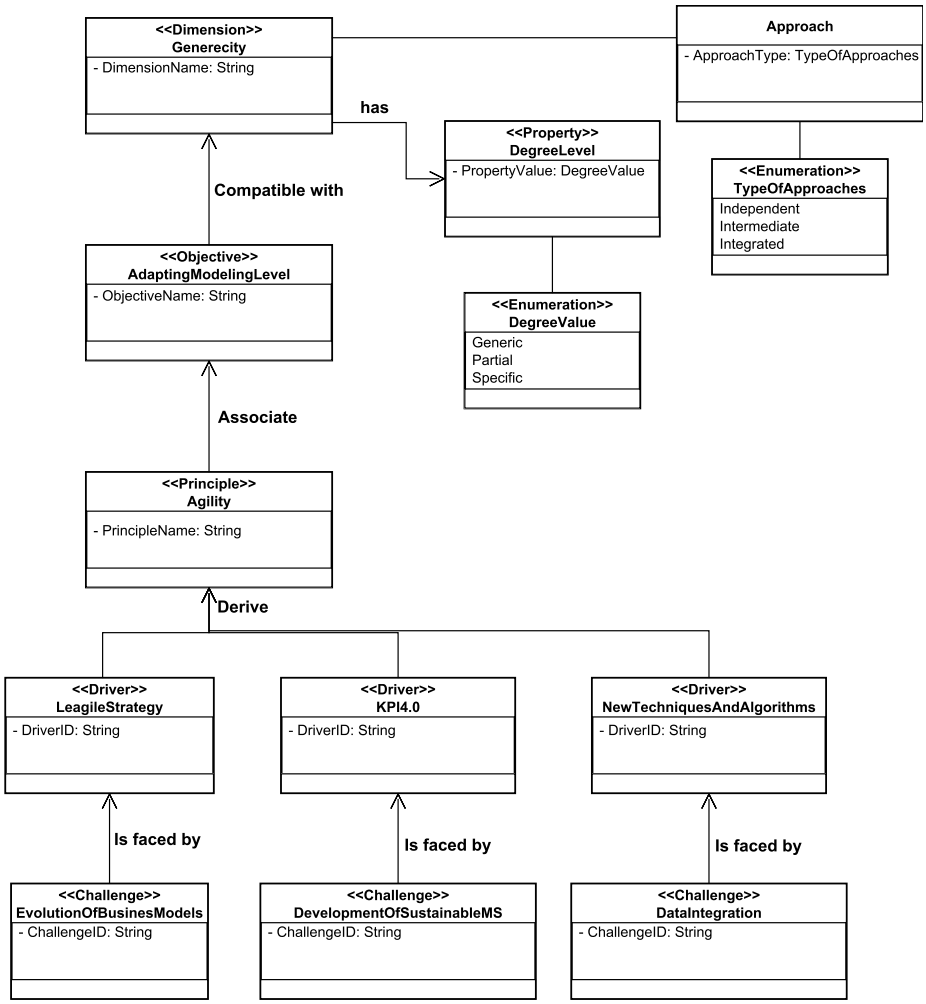

Figure C.6: Conceptual model for the genericity dimension.

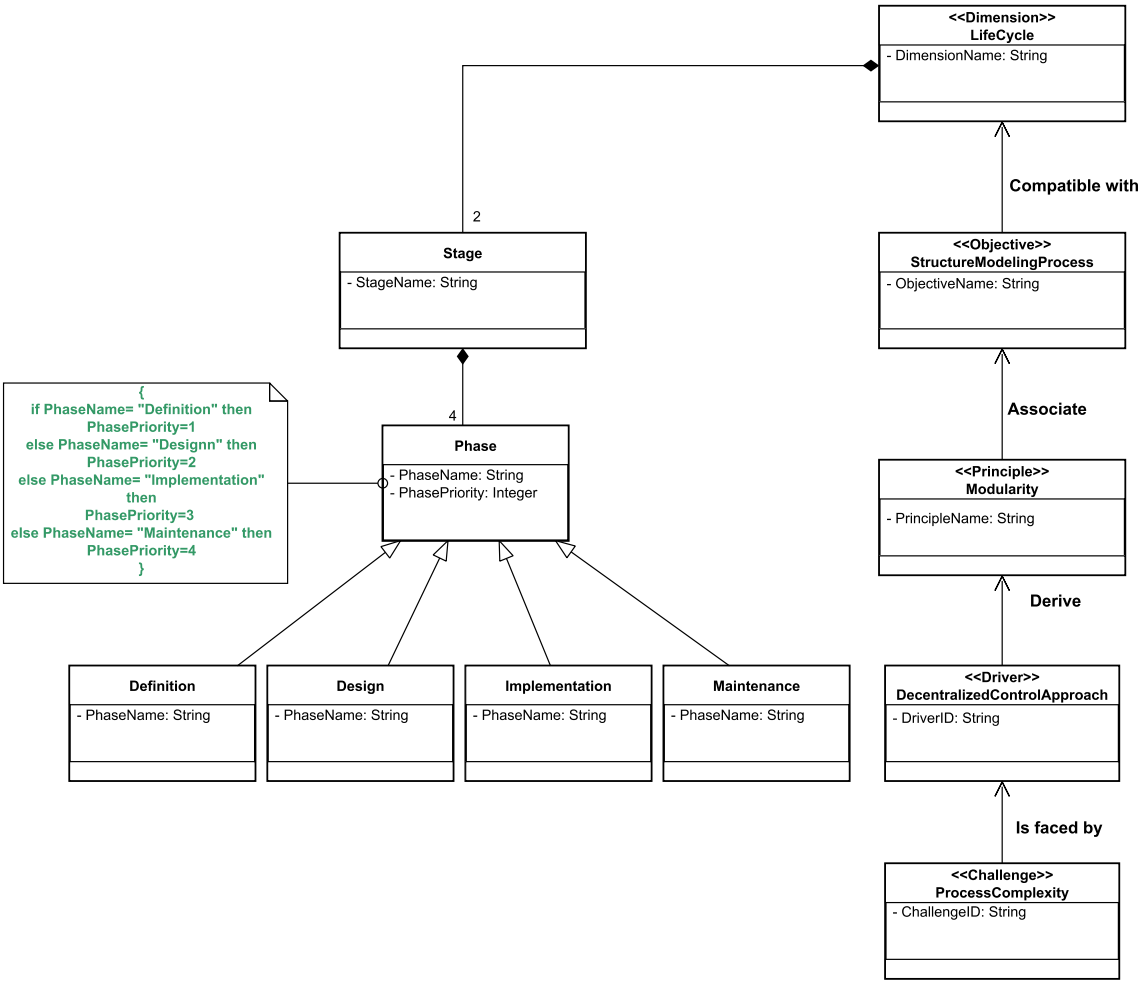

Figure C.7: Conceptual model for the life cycle dimension.

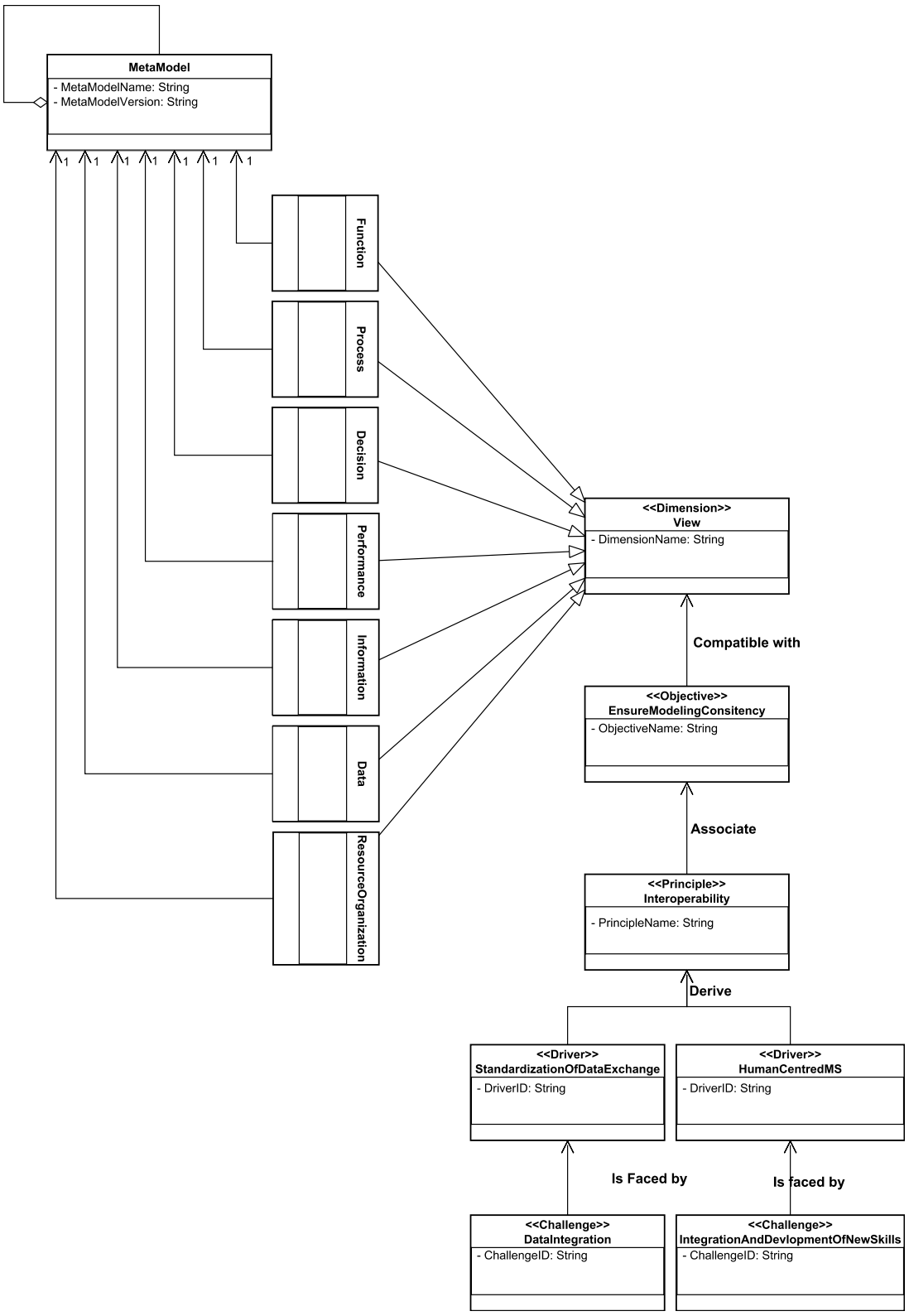

Figure C.8: Conceptual model for the view dimension.

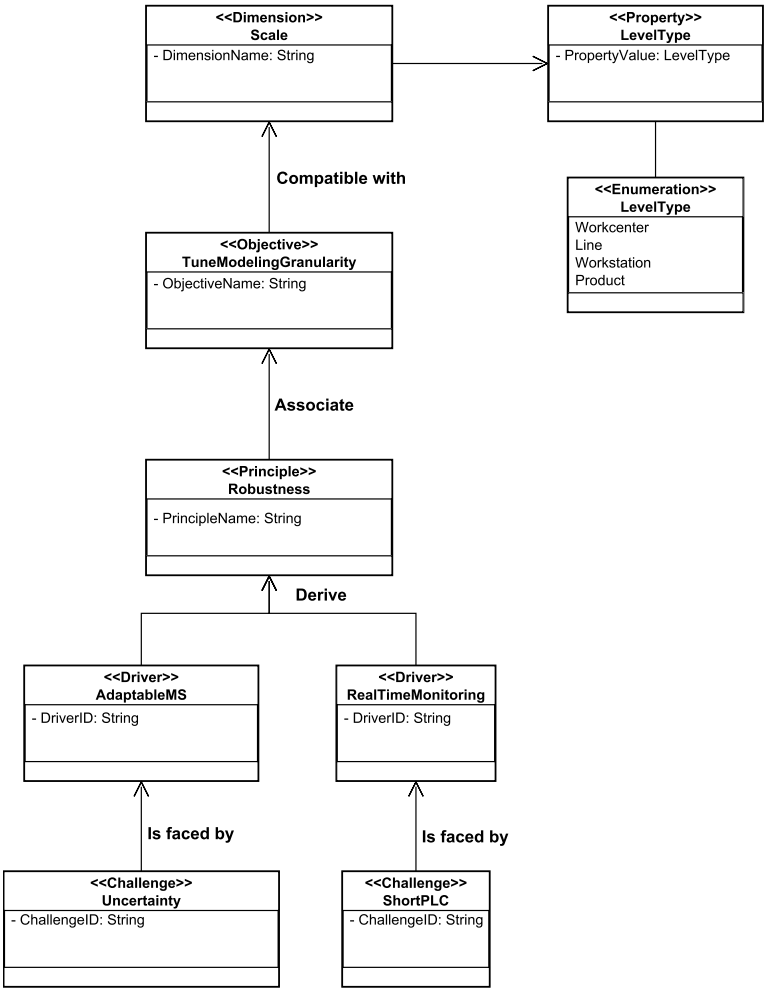

Figure C.9: Conceptual model for the scale dimension.

Appendix D

Examples of Modeling Approaches

D.1 Independent Approach: Example of Modeling the Line Feeding Policy

Let us consider the example of modeling the line feeding policy with a generic degree. If we refer to the literature, Subsection 2.3.2.2, four main policies exist, line stocking, kanban, sequencing, and kitting.

The proposed model must be able to model all these four policies, but before, let us talk about the scale level. Since the existing difference among policies is made at the product level, as illustrated by Figure D.1, the scale level is naturally imposed at a product level. In other words, in the same workstation, we could have, for example, parts stored in the supermarket X to produce P1 using the kanban policy and other parts stored in the supermarket Y to produce P2 using the line stocking.

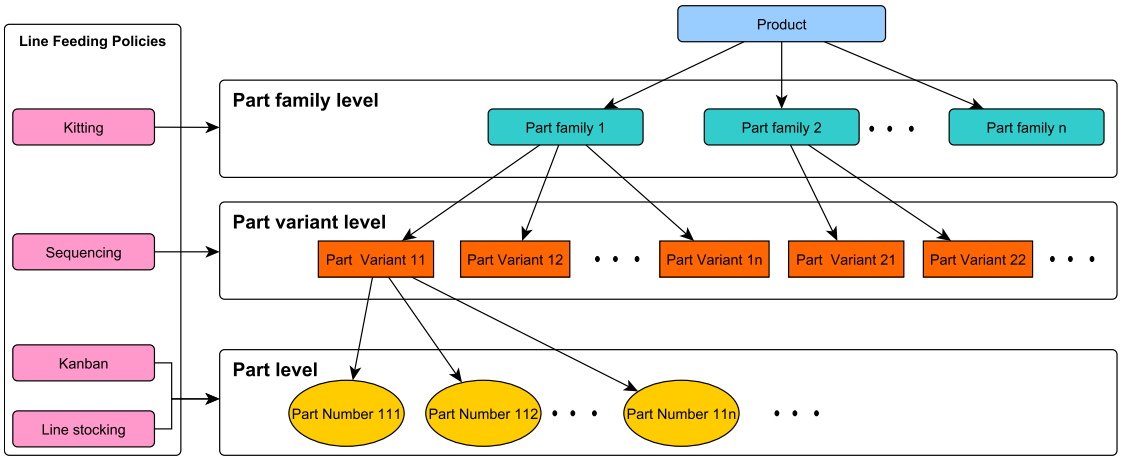

Figure D.1: The corresponding level for each line feeding policy.

If we look for common features among these policies, we will remark that they work all using a trigger. Nevertheless, conditions for generating the request event to feed the

workstations are different. More than that, for instance, in the kitting, it is possible that the workstation group triggers the component feeding requirement. In such a case, trying to find a manner to represent different policies by using a common part of the model may complexify the modeling process. Therefore, it is preferable to adopt the independent approach that preconizes to have a specific model for each policy. For instance, we use a discriminant variable (1,2,3,4) as a switch to move in the model to (kitting, sequencing, kanban, and line stocking) respectively.

D.2 Intermediate Approach: Example of Modeling the Storage Strategy

We consider the example of modeling the storage strategy with the generic degree to explain the intermediate approach. If we refer to the literature (see Subsection 2.2.3.4), two main storage strategies exist: dedicated and shared.

Figure D.2 illustrates the workflow of assigning a location to the item in the warehouse according to the two strategies. We could see that the reception of items, control, assignment to the location, and storage process are common features. In contrast, the decision of the storage location differs from the dedicated to the shared strategy. The former calls the optimization model of COI, and the second search only for an empty location. Therefore, the n-switch mechanism used for the decision of the location allows having a specific treatment depending on the storage strategy of the item.

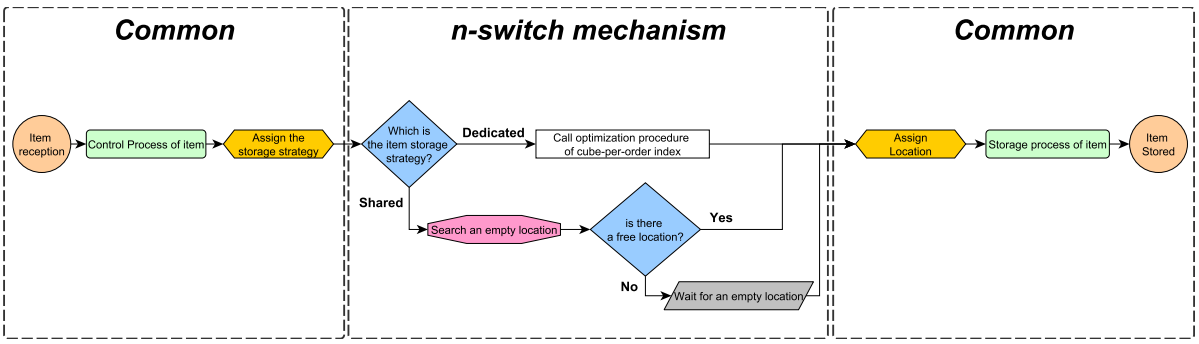

Figure D.2: Example of intermediate approach to model storage strategies.

Usually, the intermediate approach could be a good compromise when it is possible to identify common features quickly between concepts to model, and struggle to adopt the integrated approach. This later is explained hereafter.

D.3 Integrated Approach: Example of Modeling the Production Policy

An example of Figure D.3 illustrates three production policies (Push, pull, and hybrid).

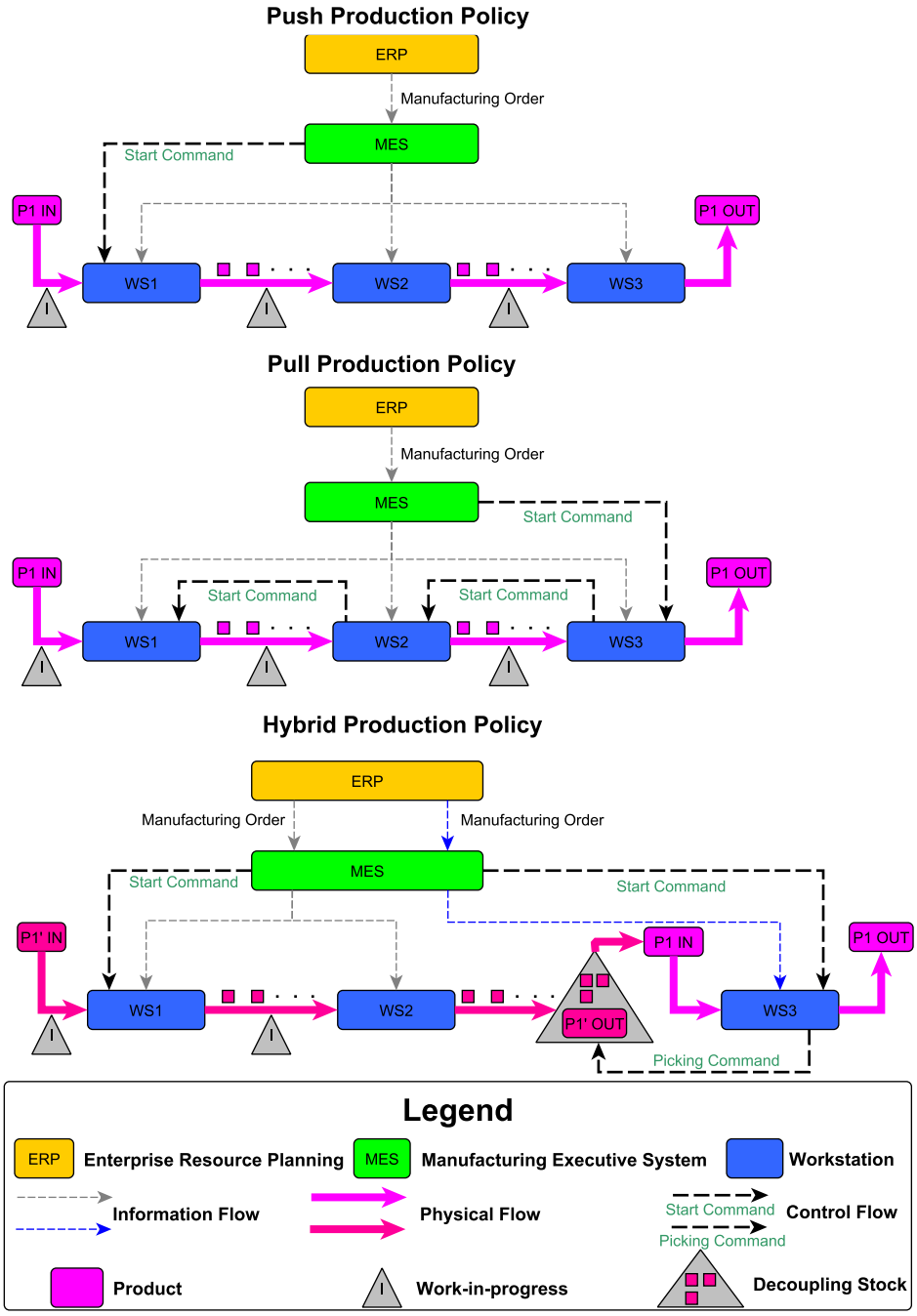

Figure D.3: Illustration of push and pull production policies. For simplification reasons, feedback from workstations to MES is omitted.

Suppose we try to model the production policy submodule (control part) with a generic

degree, i.e., able to model all possible production policies. Then, according to the integrated approach, we need to have only one model for modeling such cases.

Usually, the production policy is decided per line, but, for genericity reasons, we could consider that each workstation could have its policy. Therefore, we fix the scale at the workstation level.

Let us analyze the common and different features of the three policies. The pull policy is an upstream strategy and works using signals between workstations. In other words, a workstation could not start producing if it does not receive an order from the previous workstation. While the push policy is considered a downstream strategy. Material flow from one workstation to the next workstation triggers the start of production. The hybrid policy consists of combining the two. For instance, when dealing with the postponement strategy, we could have the two first workstations working under push policy and the latest workstation working under pull policy.

A trick to use is to see the material flow as a signal for modeling the push policy. Therefore, moving a material flow from one workstation to the next one will automatically generate a signal to trigger the production (as illustrated in Figure D.4).

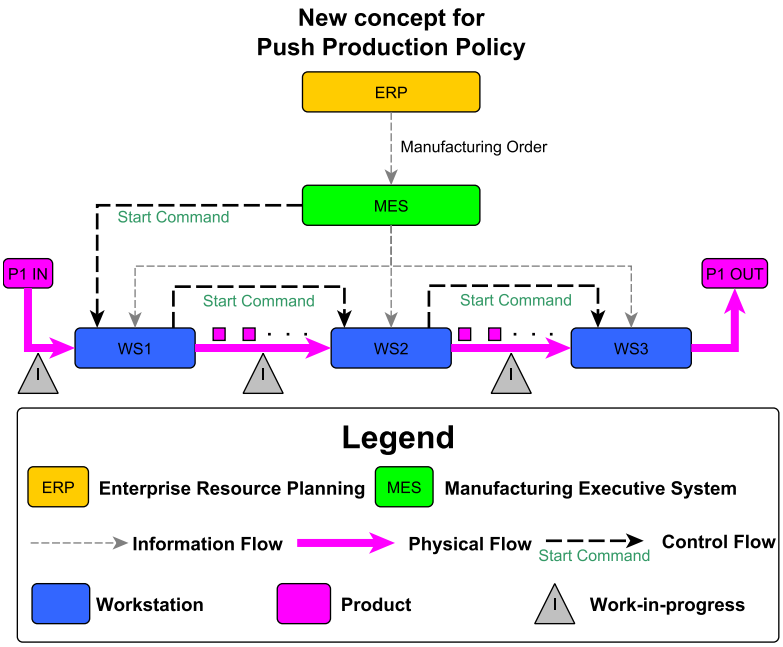

Figure D.4: Modeling the push policy using the concept of signal from a workstation to the next one.

After doing this, we could generalize the concept and say that each workstation needs to receive a signal to start the production, whatever the chosen production policy. Then, when it is finished producing, it must send a signal to the next workstation. Then using the same model, we could switch between different policies.

Appendix E

Application of MEMO I4.0: Didactic Example of Entreprise E1

E.1 Roadmap of Implemntation of MEMO I4.0 (detailed version)

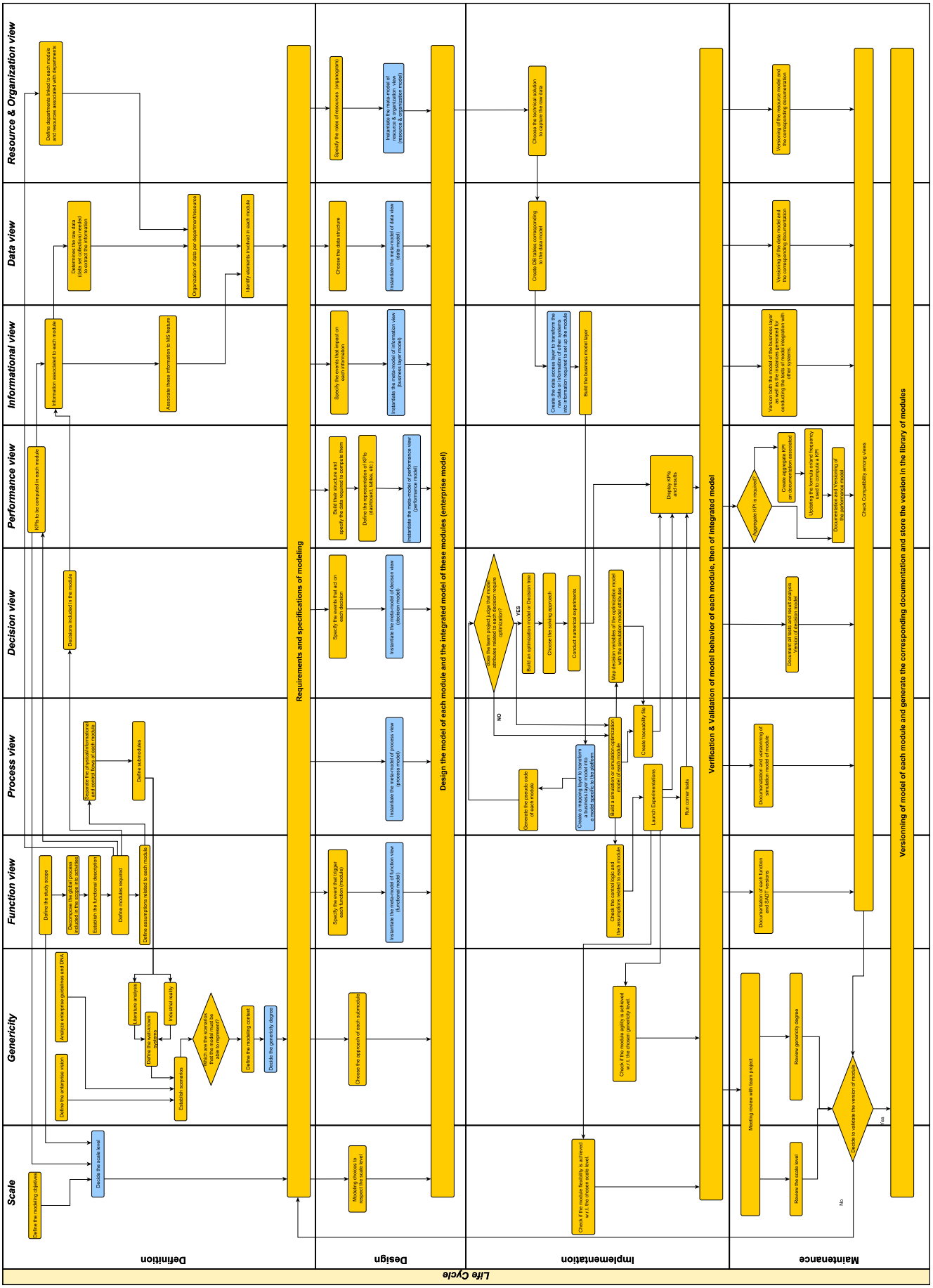

Figure E.1: Roadmap for implementing MEMO I4.0.

E.2 Definition Phase of MEMO I4.0: Didactic Example of E1

E.2.1 Assumptions related to the Modules of the Production Activity of E1

Table E.1: Assumptions related to the production activity of E1.

Module	Designation	Code
Plan	Demand patterns for pr are indep	E.a.2
	Shortage is not permitted	Z.a.4
	Multi ressource	R.a.1
	Variety of pr	Z.a.5
	Discrete time	T.a.1
	Multi-period	T.a.4
Schedule	No job spliting	Z.a.6
	No job interuption	Z.a.7
	Fixed Number of position per job	Z.a.8
	At any time, at most one job per Machine	R.a.18
	Task times are known	T.a.9
	Sequence dependent setup times	T.a.16
	Setup times are negligible	T.a.17
Assembly	Production seq is known	P.a.11
	Several pr can be manufactured (t)	P.a.16
	Cost is dep or indep from WS and tool	C.a.1

E.2.2 Choice of Scale level of Assembly Module of E1

We try to model the manufacturing order submodule of E1. Figure E.2 illustrates the case of push production policy and indicates at which level the manufacturing order, the execution order, or the transfer order could be managed.

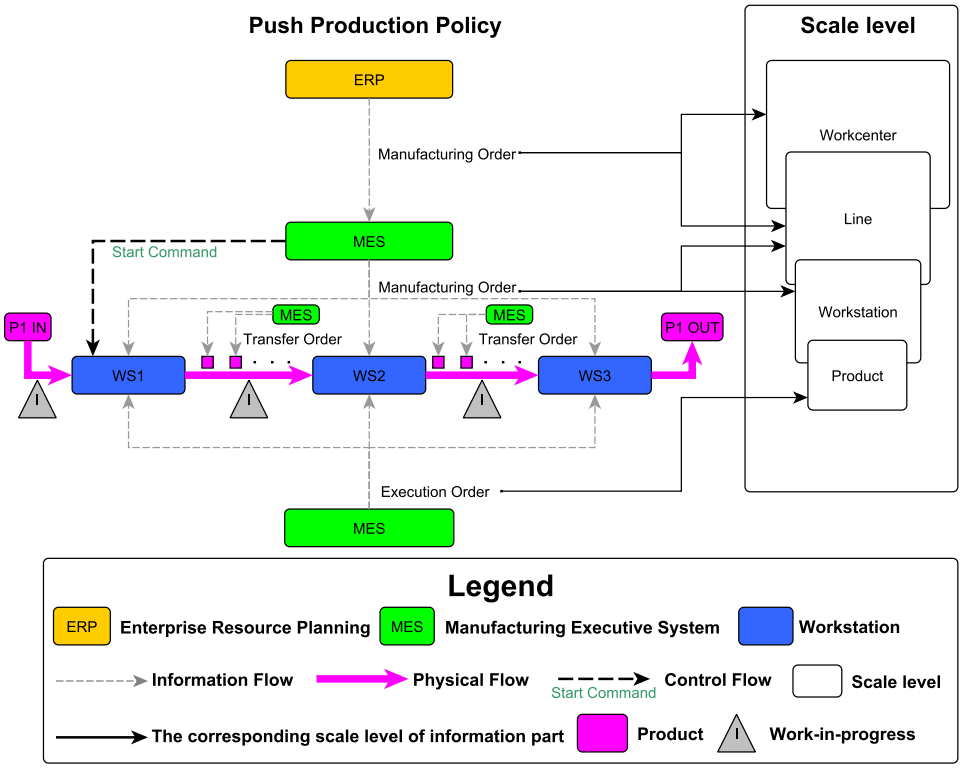

Figure E.2: Example of scale level using the manufacturing order submodule.

For instance, we could choose to model the manufacturing order at the work center level, mainly if we are interested in managing manufacturing orders in the ERP system. Because what is important to capture in the ERP is the cost linked to the execution of the manufacturing order. Usually, we prefer to manage the production cost at the work center level in industrial companies. However, if cost aggregation is not significant in some cases, choosing the line level could be more accurate. For instance, if we have dedicated lines, a part of them manufacture standard products, while others focus on only exotic products, then aggregating costs at the work center level may not reflect operational reality.

In the case of modeling the manufacturing order in the ERP, the module's parameters will be the operation times, the consumption of materials in each operation, the work center or the line where the operation will be executed, and finally, the operation sequence.

On the other hand, it will be preferable to choose the line or workstation as the scale level to model the manufacturing order in the MES system. Because, for the execution of manufacturing orders, MES needs to have an exact configuration of each work center and capabilities of each resource to assign them to the tasks that must be performed.

Suppose we need to have a real-time monitoring and live tracking of each product of the manufacturing order lot. In that case, we need to go to the product level. Therefore, we could obtain an accurate digital twin of process/product. In other words, each product must be identified by its serial number. Thus, we generate an execution order to start

the production in each workstation and a transfer order for moving a product from one workstation to another. By doing so, we can track the product during its manufacturing process in real-time.

In the case of modeling the manufacturing order in the MES, the parameters of the module will be :

- The task list of each operation and their sequence
- The task times
- The consumption of materials in each task
- The workstation where the task will be executed
- The related documents such as traceability and conformity.

E.2.3 Modeling Approach of Physical Submodule of Assembly Module of E1

Our aim is to provide the approach to adapt for modeling the physical submodule (MS structure) of the assembly module of E1, with a predetermined level of genericity intended to enable the modeling of DMS, FMS, and RMS.

For instance, at the workstation level, we consider the station, the cell, and the stage the main concepts used for structuring the production activity in DMS, FMS, and RMS, respectively. Figure E.3 illustrates an example of these three concepts.

In terms of modeling, if we choose the scale at the workstation level, the machine is an example of a common concept among the station, the cell, and the stage. However, differences in terms of layout and material handling systems exist. In our case, we decided to use the integrated approach. Then we define an abstract concept such as the configuration for dealing with the concepts of layout and material handling system, A configuration may have several workstations (stages, cells, or lines). Besides, we could have many similar resources (machines, operators, tools, fixtures, or a combination) in each workstation. On the other hand, feature and capability concepts could manage resource parameters, including set-up, setting, and failure.

In summary, choosing between independent, intermediate, or integrated approaches during the modeling remains a case-related decision. It is important to keep in mind that using an abstract concept may complexify the model, as we illustrated by the example of the line feeding policy. Therefore, it is recommended to review the choice of the approach by analyzing the common features and differences correctly.

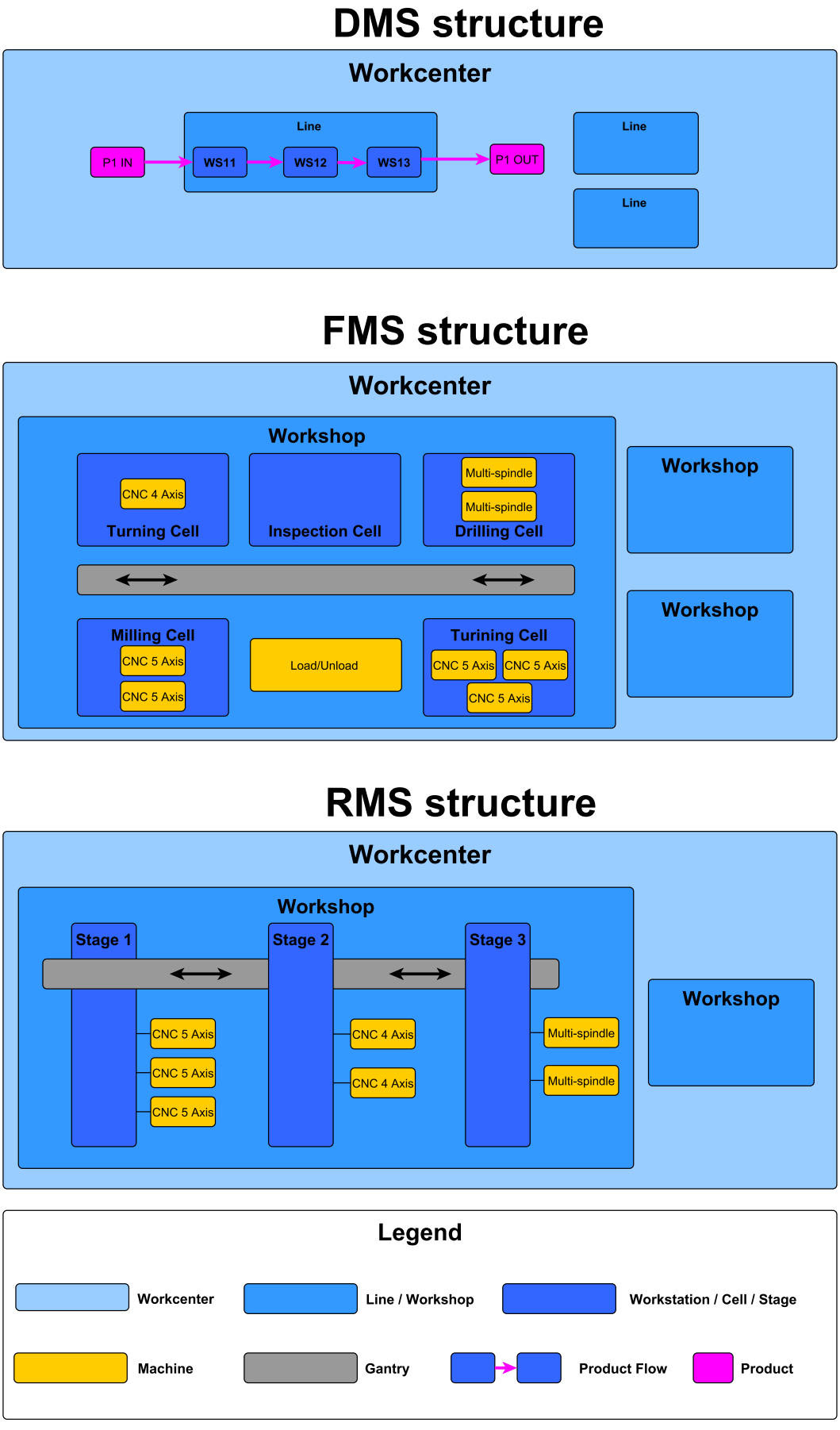

Figure E.3: Example of structure of DMS, FMS, and RMS.

E.3 Design Phase of MEMO I4.0: Didactic Example of E1

E.3.1 Modeling Choices of Each Submodule of E1 Modules

We fixed the scale level at workcenter level for the plan module, at line level (cell) for the schedule module, and at the workstation level for the assembly module. Therefore, the modeling choice of elements must be compatible with these scale levels. Columns containing "x" indicate that the element is involved in the submodule.

Table E.2: Modeling choices for submodules of E1 modules.

Modeling choice	**Module**	**Plan**		**Schedule**			**Assembly**		**All**
	Element / Submodule	*Demand control*	*Customer order*	*Planned MO*	*Shift control*	*Production control*	*Manufacturing order*	*Production process*	*Period control*
Resource	Workcenter	x							
Attributes of the entity Inventory to control the demand	Component	x							
	Period	x							
Advanced Set	BOM	x							
Attributes of the entity customer order	Workcenter		x						
	Product		x						
	Period		x						
	Demand		x						
Attributes of the entity PMO	Workcenter			x					
	Cell			x					
	Product			x					
	Period			x					
	BOM			x					
	Component			x					
Attributes of shift entity	Workcenter				x				
	Cell				x				
	Shift				x				
	Process (ShiftTime)				x				
	Period				x				
Attributes of the entity control resource capacity	Workcenter					x			
	Cell					x			
	Workstation					x			
	Period					x			
Attributes of the entity MO	Workcenter						x		
	Cell						x		
	Workstation						x		
	Product						x		
	Period						x		
	BOM						x		
	Component						x		
	Operation						x		
Attributes of the product entity	Workcenter							x	
	Cell							x	
	Workstation							x	
	Product							x	
	Process (production time)							x	
Attributes of the period entity	Workcenter								x
	Period								x
	Process (Period time)								x

E.3.2 Instantiation of the Function View Metamodel of E1

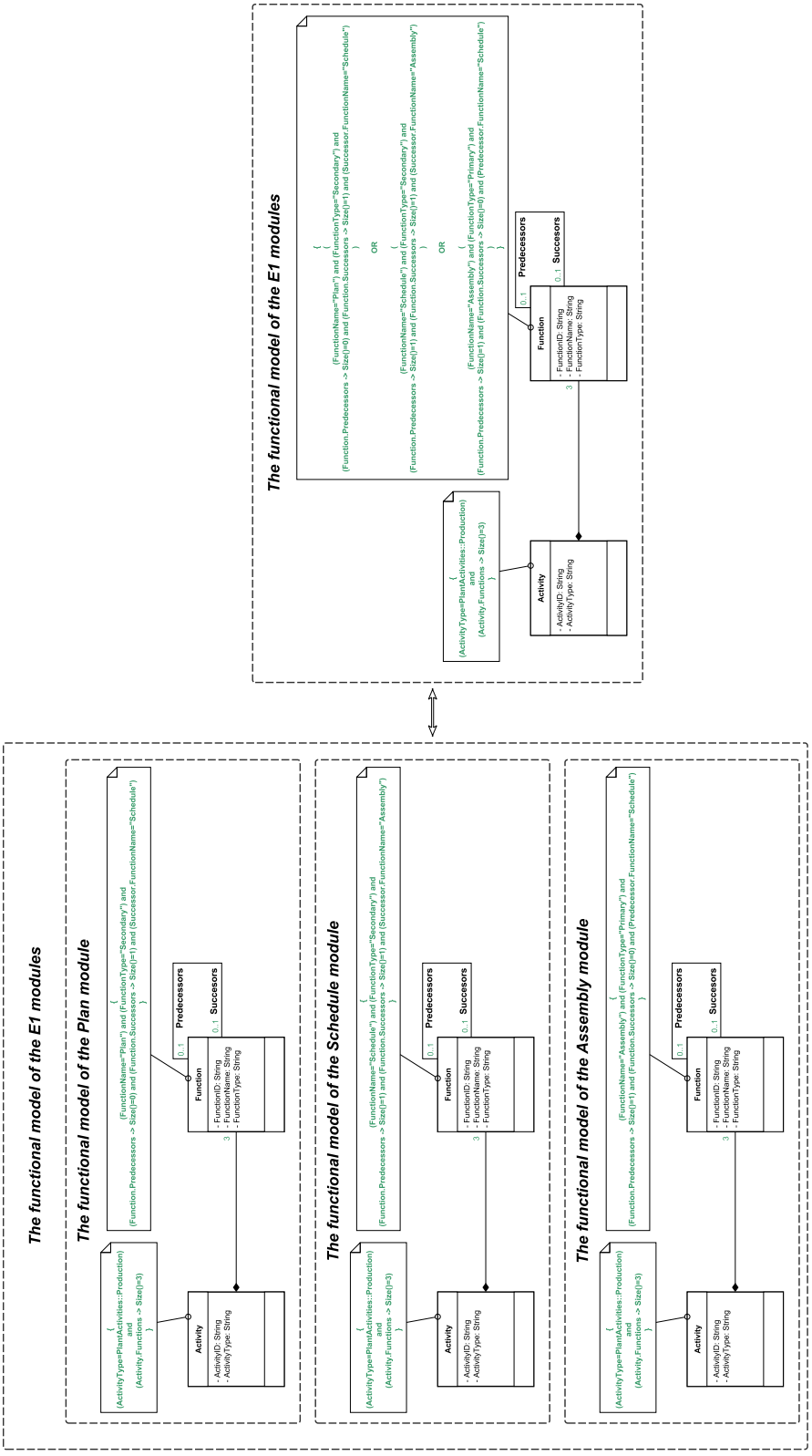

Figure E.4: Model of function view of the all modules of E1.

E.4 Implementation Phase of MEMO I4.0: Didactic Example of E1

E.4.1 Mapping of the Decision Variables of the Schedule Module of E1 with Attributes of the Simulation Model

For each period, we solve the optimization model that provides the values X_{ij}, which specify the starting time of job j in machine i. This decision variable will allow to compute the rank of each job (product), and to obtain the condition for starting the processing of product in the machine (workstation).

Table E.3: Mapping decision variables of the Schedule module of E1 with attributes of the simulation model.

Module	Variable	Data Set Collection	Function Data	MS Feature Code	Corresponding attributes in the simulation model	Mapping with modeling choices
Schedule	X_{ij}	ProducionSequencing	Output	P.p.16	aRank(Product, Period)	Rank in the queue of process element of product entity in the current period
		ProductionScheduling		T.p.10	aStartTime(Product, Period)	Time to leave the queue of process element of product entity in the current period

E.4.2 List of Notations Used in the Pseudo-Codes

This appendix defines the list of notations we use in the pseudo-codes.

Table E.4: The meaning of the notations used in the pseudo-code (Part 1).

Notation	Meaning
WC	Workcenter
P	list of products
$S(P)$	Set of the list of products
$S_t(P)$	Set of products P in period t
$H(P,T)$	Historical demand of a list product during a time horizon T
RC	Resource capacity
$BOM(P)$	Set of the Bill of materials of the list of product P
$Inv(P)$	Set of the inventory of the list of product P
$Inv(C_i)$	Set Inventory of the Component C_i
$Inv_t(C_i)$	Inventory of the Component C_i during the period t
$Inv_t(p)$	Inventory of the product p during the period t
$D_{t+1}(p)$	Demand of the product p during the period t+1
$QP(C_i)$	Quantity to be produced of C_i
$QR_{t+1}(C_i)$	Quantity required of C_i during the period t+1
$QO(C_i)$	Quantity ordered of C_i
$QP(p)$	Quantity to be produced of p
$PMO(C_i, QP(C_i))$	Planned manufacturing order of C_i with quantity QP to be produced
$PO(C_i, QO(C_i))$	Purchase order of C_i with QO to be ordered
$PMO(P, QP(P))$	Set of planned manufacturing order of P with quantity QP to be produced
$PMO(p, QP(p))$	planned manufacturing order of p with quantity QP to be produced
$Load(C_i)$	Work load when producing C_i
$LR(C_i)$	Load rate of C_i
$Load(p)$	Work load when producing p
$LR(p)$	Load rate when producing p

Table E.5: The meaning of the notations used in the pseudo-code (Part 2).

Notation	Meaning
Π	The planning of production of manufacturing order
MO	A manufacturing order
$OPSeq$	An Operating sequence
QP	A quantity to be produced
$MO(C_i, QP(C_i))$	Manufacturing order of C_i with quantity QP
$MO(p, QP(p))$	Manufacturing order of p with quantity QP
OP_i^1	First Operation of the operating sequence
OP_i	Operation i of of the operating sequence
$QP(MO)$	Quantity to be produced in a manufacturing order
$OPSeq(MO)$	Operating sequence of manufacturing order
$Q(OP_i)$	Quantity produced in operation i of the manufacturing order
R_{ij}	Matrix of assignement of resource j to the operation i
$< ComponentList_i, Q_i) >$	Set of component list and their quantity to be consumed when executing the operation i
Q_{ik}	Quantity to be consumed of component k during the execution of operation i
C_{ik}	Component k of the $ComponentList_i$
$Inv(C_{ik})$	Inventory of component k of the $ComponentList_i$
$F_t(d)$	Function of distribution of the demand during a period t
$\hat{F}_{t+1}(d)$	Estimated function of distribution of the demand d
μ	Constante
σ	Constante
$\hat{\mu}$	Estimated value of μ
$\hat{\sigma}$	Estimated value of σ
α, β	Constante

E.4.3 List of Algorithms (the Pseudo-Codes)

This appendix defines the list of algorithms in form of pseudo-codes that we use in the process view.
Below, we provide the pseudo-codes corresponding to the functions of the process. For the meanings of the notations used in the pseudo-code, we invite the reader to refer to the Tables E.4 and E.5.

Algorithm 2 Pseudo-Code of Plan Function.

Input: $WC, S(P), H(P,T), RC, BOM(P), Inv(P), Inv(C_i)/C_i \in BOM(P)$
Output: $PMO(C_i, QP(C_i)), PO(C_i, QO(C_i)), PMO(P, QP(P))$

Require: *Time = Planning period Or New customer order*
Ensure: *Meet all customer orders*
$m \leftarrow getSize(P, WC)$ ▷ Number of products in the workcenter (WC)
for $p \leftarrow 1, m$ **do**
  $D_{t+1}(p) \leftarrow$ Forecast$(S(p), H(p,T))$ ▷ Call the Forecast Algorithm
  $n \leftarrow getSizeOfComponentList(p)$
  for $i \leftarrow 1, n$ **do**
    if $C_i \in BOM(p)$ **then**
      $QR_{t+1}(C_i) \leftarrow$ Compute required quantity of C_i in t+1
      if $QR_{t+1}(C_i) > Inv_t(C_i)$ **then**
        if isManufactured(C_i) **then**
          $PMO(C_i, QP(C_i)) \leftarrow$ Create a Planned Manufacturing Order of C_i
          $PMO(C_i, QP(C_i))$ *status* $\leftarrow$ "Create"
          $Load(C_i) \leftarrow$ Estimate work load using $PMO(C_i, QP(C_i))$
          $LR(C_i) \leftarrow \frac{Load(C_i)}{RC}$ ▷ Compute load rate
          if $LR(C_i) > 100\%$ **then**
            Perfom the resource leveling
          end if
          Release $PMO(C_i, QP(C_i))$
          $PMO(C_i, QP(C_i))$ *status* $\leftarrow$ "Release"
        else
          $PO(C_i, QO(C_i)) \leftarrow$ Create a purchase order of C_i
          $PO(C_i, QO(C_i))$ *status* $\leftarrow$ "Create"
          release $PO(C_i, QO(C_i))$
          $PO(C_i, QO(C_i))$ *status* $\leftarrow$ "Release"
        end if
      end if
    end if
  end for
  if $D_{t+1}(p) > Inv_t(p)$ **then**
    $PMO(p, QP(p)) \leftarrow$ Create a Planned Manufacturing Order of p
    $PMO(p, QP(p))$ *status* $\leftarrow$ "Create"
  end if
  $Load(p) \leftarrow$ Estimate work load using $PMO(p, QP(p))$
  $LR(p) \leftarrow \frac{Load(p)}{RC}$ ▷ Compute load rate
  if $LR(p) > 100\%$ **then**
    Perfom the resource leveling
  end if
  release $PMO(p, QP(p))$
  $PMO(p, QP(p))$ *status* $\leftarrow$ "Release"
end for

Algorithm 3 Pseudo-Code of Schedule Function.

Input: $WC, Productioncalendar, t, PMO(P, QP(P)), PMO(C_i, QP(C_i))/C_i \in BOM(P)$

Output: $\Pi, MO(C_i, QP(C_i)), MO(P, QP(P))$

Require: $New\ planned\ manufacturing\ order, Periodt + 1$
Ensure: $Confirmation\ of\ planned\ manufacturing\ order$

1: **for all** $PMO\&\&MOofperiodt + 1$ **do** $nbCells \leftarrow getNumberOfCells(WC)$ ▷ Number of cells in the workcenter WC $m \leftarrow getSize(P, WC)$
2: **for** $p \leftarrow 1, m$ **do**
3: $n \leftarrow getSizeOfComponentList(p)$
4: **for** $i \leftarrow 1, n$ **do**
5: $PMO(C_i, QP(C_i))$ $status \leftarrow$ "End"
6: $MO(C_i, QP(C_i)) \leftarrow$ Create a Manufacturing Order of C_i
7: $MO(C_i, QP(C_i))$ $Status \leftarrow$ "Create"
8: **end for**
9: $PMO(p, QP(p))$ $status \leftarrow$ "End"
10: $MO(p, QP(p)) \leftarrow$ Create a Manufacturing Order of p
11: $MO(p, QP(p))$ $Status \leftarrow$ "Create"
12: **end for**
13: $\Pi \leftarrow$ Elaborate the sequencing & scheduling of manufacturing orders according to the production calendar and shifts of period $t + 1$, and cells in the workcenter WC
14: **for** $p \leftarrow 1, m$ **do**
15: **for** $i \leftarrow 1, n$ **do**
16: $MO(C_i, QP(C_i))$ $Status \leftarrow$ "Scheduled"
17: **end for**
18: $MO(p, QP(p))$ $Status \leftarrow$ "Scheduled"
19: **end for**
20: **end for**

Algorithm 4 Pseudo-Code of Produce Function.

Input: $WC, MO, OPSeq$
Output: MO *status*

Require: $New\ manufacturing\ order, Periodt + 1$
Ensure: *All MO status are End*
$nbCells \leftarrow getNumberOfCells(WC)$ ▷ Number of cells in the workcenter WC
for $cell \leftarrow 1, nbCells$ **do** $nbWS \leftarrow getNumberOfWorkstations(cell)$ ▷ Number of workstations in the cell
  for $ws \leftarrow 1, nbWS$ **do**
    Check the availability of all components in the ws
    Check the availability of resources in the ws
  end for
end for
for $MO \in MO(C_i, QP(C_i)) \cup MO(P, QP(P))ofperiodt + 1$ **do** ▷ Manufatcuring Orders of period $t + 1$
  $MO\ \ Status \leftarrow$'Release'
  $OP_1 \leftarrow getFirstOperation(OpSeq(MO))$
  $QP \leftarrow QP(MO)$
  $OPSeq \leftarrow OpSeq(MO)$
  ExecuteOperation$(OP_1, QP, OPSeq)$
  $MO\ \ Status \leftarrow$'End'
end for
procedure ExecuteOperation$(OP_i, QP, OPSeq)$
  while $Q(OP_i) < QP$ **do**
    $R_{ij} \leftarrow 1$ Allocate resource j *to* OP_i
    if OP_i require Materials **then**
      $< ComponentList_i, Q_i > \leftarrow$ getMaterialList(OP_i)
      $n \leftarrow getSizeOfComponentList(ComponentList_i)$
      for $k \leftarrow 1, n$ **do**
        Consume $Q_{ik}\ of\ C_{ik}\ \ / C_{ik} \in ComponentList_i$
        $Inv(C_{ik}) \leftarrow Inv(C_{ik}) - Q_{ik}$
      end for
      Execute OP_i
      $Q(OP_i) \leftarrow Q(OP_i) + 1$
      if $getNext(OP_i) \neq \emptyset$ **then**
        $OP_l \leftarrow getNext(OP_i)$
        ExecuteOperation$(OP_l, QP, OPSeq)$
      end if
    end if
  end while
end procedure

Algorithm 5 Demand forecasting algorithm.

procedure FORECAST($S(P), H(P,T)$) ▷ Product set, historical demand
  if $t \in \mathbb{N}$ **then** ▷ t is discret
    $\hat{F}_{t+1}(d) \leftarrow \hat{F}_t(d)$
    if $H(P,T) \sim \mathcal{G}(\mu, \sigma^2)$ **then** ▷ General distribution
      $\hat{F}_{t+1}(d) \leftarrow G(\hat{\mu}, \hat{\sigma^2})$
    else if $H(P,T) \sim \mathcal{N}(\mu, \sigma^2)$ **then** ▷ Normal distribution
      if $\forall t \in \{0, 1, \cdots, T\}, F_t(d) > 0$ **then**
        $\hat{F}_{t+1}(d) \sim LogNormal(\hat{\mu}, \hat{\sigma}^2)$ ▷ LogNormal distribution
      else
        $\hat{F}_{t+1}(d) \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$
      end if
    else
      if $\exists\ p_2 \in S_{t+1}(P) / \forall\ p_1 \in S_t(P), p_2 \neq p_1$ **then** ▷ New products in t+1
        $\hat{F}_{t+1}(d) \sim BetaBin(N, \alpha, \beta)$ ▷ Beta binomial distribution
      else
        if *Stead Demand* **then**
          $\hat{F}_{t+1}(d) \leftarrow \alpha F_t(d) + (1-\alpha)\hat{F}_t(d)$ ▷ Simple smoothing exponentiel
        else if $Trend$ **then**
          $\hat{F}_{t+1}(d) \leftarrow 2F_t(d) - F_{t-1}(d) - 2(1-\alpha)e_t + ((1-\alpha)^2 e_{t-1})/e_t = F_t(d) - F_{t-1}(d)$ ▷ Double smoothing exponentiel
        else ▷ Seasonal demand
          $\hat{F}_{t+1}(d) \leftarrow Use\ triple\ smoothing\ exponentiel$
        end if
      end if
    end if
  else
    $\hat{F}_{t+1}(d) \sim Geometric\ Brownian\ motion\ distribution$
  end if
  return $\hat{F}_{t+1}(d)$
end procedure

Appendix F

Application of MEMO I4.0: elm.leblanc company

F.1 Definition Phase

F.1.1 List of the Modeling Scenarios of Modules of the Project MEMO I4.0 within elm.leblanc

This appendix defines the modeling scenarios related to the submodules of the prepare, load, transport and produce modules. We defined them in the project MEMO I4.0 within elm.leblanc company to specify the right genericity degree of each submodule.

We use the green color in the background to show the evolution from the current situation to the future one, including new capabilities. In comparison, the gray color in the background illustrates no changes between current and future situations. Furthermore, the bold red color of the font means that is the current situation of the elm.leblanc company. Finally, the orange color of the background with "X" indicates that it is a deprecated situation, thus, not envisaged.

Table F.1: Scenarios to consider for modeling the preparation process submodule.

Current \Future preparation time	Independant from the storage support	Dependant from the storage support
Independant from the storage support	1	2
Dependant from the storage support	X	**3**
Current / Future storage support	Only one type	Multiple types
Only one type	1	2
Multiple types	X	**3**
Current / Future # of the storage area	Fixed	Variable
Fixed	**1**	2
Variable	X	3

Table F.2: Scenarios to consider for modeling the preparation order submodule.

Current \Future # of components per preparation order	Single component	Multiple Components
Single component	1	2
Multiple Components	X	3
Current / Future # storage area per preparation order	Single storage area	Multiple storage areas
Single storage area	1	2
Multiple storage areas	X	3

Table F.3: Scenarios to consider for modeling the preparation control submodule.

Current \Future Line feeding Policy	Line stocking	Kanban	JIS	Kitting
Line stocking	1	2	3	4
Kanban	X	5	6	7
JIS	X	8	9	10
Kitting	X	11	12	13
Cuurent / Future # of assigned resources to the picking list	Single preparator		Multiple preparators	
Single preparator	1		2	
Multiple preparators	X		3	

Table F.4: Scenarios to consider for modeling the shift/resource control submodule.

Current \Future capacity of resources /Shift/Date	Static	Dynamic
Static	1	2
Dynamic	X	3

Table F.5: Scenarios to consider for modeling the loading process submodule.

Current \Future Cart Type	Single type	Multiple type
Single type	1	2
Multiple type	X	3
Current / Future # of cart stages	One stage /Cart	Multiple stages /Cart
One stage /Cart	1	2
Multiple stages /Cart	X	3
Current / Future capacity of cart stages	Identical	Different
Identical	1	2
Different	X	3

Table F.6: Scenarios to consider for modeling the transport process submodule.

Current \Future # of transporter type	One Type	Several type
One type	**1**	2
Several type	X	3
Current / Future organization of picking station in terms of line	Dedicated to one line	Shraed between lines
Dedicated to one line	1	2
Shared between lines	X	**3**
Current / Future organization of picking station in terms of cart type	Dedicated to one cart type	Dedicated to several cart type
Dedicated to one cart type	**1**	2
Dedicated to several cart type	X	3
Current / Future capacity of picking station	Similar for all lines	Different between lines
Similar capacity for all lines	**1**	2
Capacity different for each line	X	3

Table F.7: Scenarios to consider for modeling the picking order submodule.

Current \Future picking policy	Single order picking	Batch picking	Zone picking	Wave picking	Backet brigade picking
Single order picking	1	2	3	4	5
Batch picking	X	6	7	8	9
Zone picking	12	13	14	15	16
Wave picking	18	19	20	21	22
backet brigade picking	24	25	26	27	28

Table F.8: Scenarios to consider for modeling the transport control submodule.

Current \Future # carts per picking order	Single cart	Multiple carts
Single cart	1	2
Multiple carts	X	**3**
Current / Future # of storage area	Single storage area	Multiple storage area
Single storage area	**1**	2
Multiple storage area	X	3

Table F.9: Scenarios to consider for modeling the production process submodule.

Cuurent \Future MS type	DMS	FMS	RMS
DMS	1	2	3
FMS	X	4	5
RMS	X	X	6
Current / Future # of products per line	Single	Mix product	
Single	1	2	
Mix product	X	3	
Current / Future # of supermarkets feeding one station	One supermarket	Several supermarkets	
One supermarket	1	2	
Several supermarkets	X	3	

Table F.10: Scenarios to consider for modeling the manufacturing order submodule.

Current \Future # of products per manufacturing order	Single product	Multiple product
Single product	1	2
Multiple product	X	3
Current / Future # of lines per manufacturing order	Single Line	Multiple Line
Single Line	1	2
Multiple Line	X	3

Table F.11: Scenarios to consider for modeling the production control submodule.

Current \Future production policy		Push	Pull	Hybrid
Current / Future organization of production process	Push	1	2	3
	Pull	X	4	5
	Hybrid	X	X	6
Current / Future # of assigned production resources to operation		One piece flow	Batch process	Multiple resource
	One piece flow	1	2	
	Batch process	X	3	
	Single resource	1	2	
	Multiple resource	X	3	

F.1.2 List of Tables related to the Function Data of Modules

This appendix defines the list of tables related to the function data of modules defined in the project MEMO I4.0 within elm.leblanc company. It includes Information, dataset collection. Note that we consider the rows colored with a blue color as a setting of MS within elm.leblanc, while the row with a green color refer to a new MS feature that must be created in the CT-PPRE table.

Table F.12: Function data of the Prepare module and their associated MS features.

Module	Function Data	Information	MS feature	MS Feature Name
Prepare	PreparationParameters	Process (Preparation) time / Storage support	T.w.7	Item picking time
		Maximum number of preparation orders / L	P.s.8	Preparation batch size
		Waiting time before printing PRO / L	Setting	Setting
		Size of shift of preparation in storage area / L	R.c.17	# of pickers/zone
		Option of selecting preparatos	Setting	Setting
		Option of preparation	Setting	Setting
		Number of attributes / L / P	Setting	Setting
		Number of rows / L / W	Setting	Setting
	PreparatorsData	Preparator list in the storage area / L / Date / Shift	R.w.22	Assignment of pickers to zones
	InventoryData	Initial Inventory of component / L	Z.α.9	Current line-side inventory
		Flag of launching preparation oder	Setting	Setting
	ComponentsData	Component reference / L / W / P	Z.II.4	Set of comp
		Unit quantity / L / W / P	Z.c.1	Characteristics of pr
		Supply Type / L / W / P	P.s.4	Kanban, L Stocking , Sequencing, Kitting
		Supermarket / L / W / P	P.s.6	Parts each SUP should handle
		Supermarket capacity / Component support / L / W / P	R.c.6	Cap of Sup
		Storage support / L / W / P	R.II.5	Set of pallets
			R.II.6	Set of bins
		Storage capacity / L / W / P	P.s.7	# of parts/ comp box
		Unit number of box / L / W / P	R.s.1	# of bins/comp/L
		Component support length / L / W / P		
		Component support width / L / W / P	R.w.8	Size of pick bins
		Component support height / L / W / P		
		Storage Area of component / L / W / P	P.w.20	Info on the storage area and loc

Table F.13: Data set collection, and function data of the Load module and their associated MS features.

Module	Department	Source	Data Set	Function Data	Information	MS feature	MS Feature Name
Load	Logistic	SpreadSheet	LoadingDataSet	CartsData	Capacity of carts in terms of boxes	R.c.9	Cap of each stage
					Number of stage / Cart type	R.Π.7	Set of stages/cart
					Prefered number of cart left / Line	Setting	Setting
					Number of space unit/ Cart / Line	R.c.7	Vehicle load cap
					Type of cart / Picking station	R.w.16	Transport device
	HSE	SpreadSheet	PickingStationDataSet		Space unit of empty cart / Picking station	R.c.25	Cap of each zone
					Space unit of full cart / Picking station		
	Logistic	SpreadSheet	PreparationDataSet	PreparatorsData	Preparator list in storage area / Line / Date / Shift	R.w.22	Assignment of pickers to zones
	Logistic	SpreadSheet	LoadingPreference	LoadingParameters	Preference loading box / Line	Setting	Setting
	SI	Message	LoadingTime		Process (Loading) time / Storage support	T.s.3	SKU loading time

Table F.14: Data set collection, and function data of the Transport module and their associated MS features.

Module	Department	Source	Data Set	Function Data	Information	MS feature	MS Feature Name
Transport	Logistic	SpreadSheet	TransportDataSet	TransportersData	Number of milkruns / Line	R.w.17	# of Vehicles
					Milkruns list / Line / Date / Shift	R.c.29	Carrier characteristics
				TransportParameters	Size of shift of transporters / Line	R.s.4	# of drivers/delivery
					Option of selecting transporters / Line	Setting	Setting
					Option of transport / Line	Setting	Setting
					Number of picking stations / Line	R.w.24	# of P&D WS
	SI	Message	UnlodingTime		Process (Unloading) time / Storage support	T.s.4	SKU unloading time
	Logistic	SpreadSheet	TransporterSequences		Sequence of transporters / Line	P.s.2	Optimal driver routes
	Logistic	SpreadSheet	DriversConfiguration	DriversData	Transport type / Transporter	R.w.15	MHE type
					Unit number / Transporter	R.II.8	Set of carriers
					Initial position of transporter	R.s.5	Assignment of driver to WS
					First station of feeding	Setting	Setting
					Capacity of transporter	R.c.7	Vehicle load cap
					Waiting time of transporter	Setting	Setting
					Loading time / Picking station	Setting	Setting
	HSE	SpreadSheet	DriversVelocity		Velocity for empty cart / Transporter	R.w.14	Avg speed of vehicle
					Velocity for full cart / Transporter		
	Logistic	SpreadSheet	FacilityLayout	LayoutData	Picking stations	P.w.20	Info on the storage area and loc
					Supermarkets	R.s.3	Sup Loc
					Distances among stations	P.p.13	Dist btw loc
					Networks	P.β.12	Zone config
	Process planning& maintenance	SpreadSheet	LineLayout		Workstations	R.p.6	Facility loc (t)
						R.p.7	Machine loc (t)

Table F.15: Data set collection, and function data of the Produce module and their associated MS features.

Module	Department	Source	Data Set	Function Data	Information	MS feature	MS Feature Name
Produce	Manufacturing	MAC-PAC	ProductionDataSet	ProductionParameters	Number of operations / Line	P.II.1	Set of op
					Max number of product to sequence / Line	Z.II.1	Set of pr
				ProductionPlanning	Quantity to produce / Line / Product / Date	P.p.17	Production vol
				CumulativeProductionLot	Lot quantity to produce / Line / Product / Date	P.p.22	# of pr to be produced
				LotSize	Lot size / Line / product	P.p.18	Lot size
				ProductionSequence	Sequence of production / Line / Date	P.p.16	Production seq
	Process planning& Maintenance	GASS	AssemblyDataSet	AssemblyData	Operating sequence / Line / Product	P.c.6	Precedence constraints btw tasks
					Workstation list of each operation / Line	R.c.15	Capable M/op
				AssemblyTimes	Processing time of operation / Line / Workstation / Product	T.p.2	Task time
					Setup time / Line / Workstation / Product	T.c.11	Setup-time of M
	Quality	MAS	Rework	ReworkParameters	Rework tIme / Line / Workstation / Product	Setting	Setting
					Percentage of rework / Line / Workstation / Product	Setting	Setting
					Percentage of waste / Line / Workstation / Product	Setting	Setting
	Process planning& Maintenance	Spreadsheet	ResourcesConfiguration	ResourcesData	Capacity of resource / Line	R.c.3	Max cap of L/pr
					MTBF of resource / Line	T.p.21	Mean-time btw failures
					MTTR of resource / Line	T.p.22	Mean-time to repair
	Manufacturing	MAC-PAC	ResourcesPlanning		Avalability of resources / Line / Date / Shift / Resource	R.p.11	Hired wrk (t)
						R.p.12	Laid off wrk (t)
						R.c.5	Cap available for M (t)
					Assignements of resource / Line / Date / Shift / Workstation	P.p.26	Allocation of available M cap to op
						P.p.24	Production plan/shift
	Manufacturing	Spreadsheet	ProductionOrganization	ProductionPolicies	Production and transfer policies / Line / Workstation / Product	P.g.3	Strategy of assembly
						P.p.1	MTS, MTO, ATO, ETO
					Pull table / Line / Workstation / Product	Setting	Setting
	Engineering	SAP	ProductConfiguration	ProductData	Product type / Line / Product	P.p.21	Allocation of pr to L (t)
						Z.II.2	Set of pr types
	HSE	EAGLE	ErgonomicAssessement	ErgonomicData	Operation workloads / Line	New MS feature	To be created in Process

Table F.16: Data set collection, and function data of the Common Object module and their associated MS features.

Module	Department	Source	Data Set	Function Data	Information	MS feature	MS Feature Name
CommonObject	Direction	Spreadsheet	MSConfiguration	GeneralData	Number of workstation / Line	R.g.3	# of WS/L
					Number of supermakets / Line	R.g.6	# of Sup
					Number of products / Line	Z.g.1	# of different pr
					Number of shift / Line	Setting	Setting
					Nombre of appliances / Line	Z.c.2	Max qty of pr produced/L
					Availability time / Line	T.g.4	Daily working time
					Takt time / Line	T.c.2	TT/L
					General lot size / Line	Setting	Setting
					Option dynamic or static of shift selection	Setting	Setting
	SI	Spreadsheet	SimulationSettings	SimulationData	Number replications	Setting	Setting
					Number of day / Replication	Setting	Setting
					Number of hours/Day	Setting	Setting
	Manufacturing	Spreadsheet	ShiftScheduling	ScheduleOfShiftData	Value	Setting	Setting
					Durations (min)	Setting	Setting
					IsPresent/ Shift/ Date	Setting	Setting

F.1.3 Identification of the Raw Data and its Organization per Department

Table F.17: Data set of the Prepare module, their data sources and departments ensuring their management.

Function Data	Data Set	Source	Department
PreparationParameters	Preparation time	Message	SI
	PreparationDataset	SpreadSheet	Logistic
PreparatorsData			
InventoryData	Inventory	MAC-PAC	
ComponentsData	BOM	SAP	Engineering
	ComponentFeatures	SpreadSheet	Logistic
	StorageSupportDimensions	SpreadSheet	HSE

F.2 Design Phase

Below, we provide the list of some metamodels defined in MEMO I4.0 instantiated for the Prepare module to obtain models used in the application of MEMO I4.0 within elm.leblanc company.

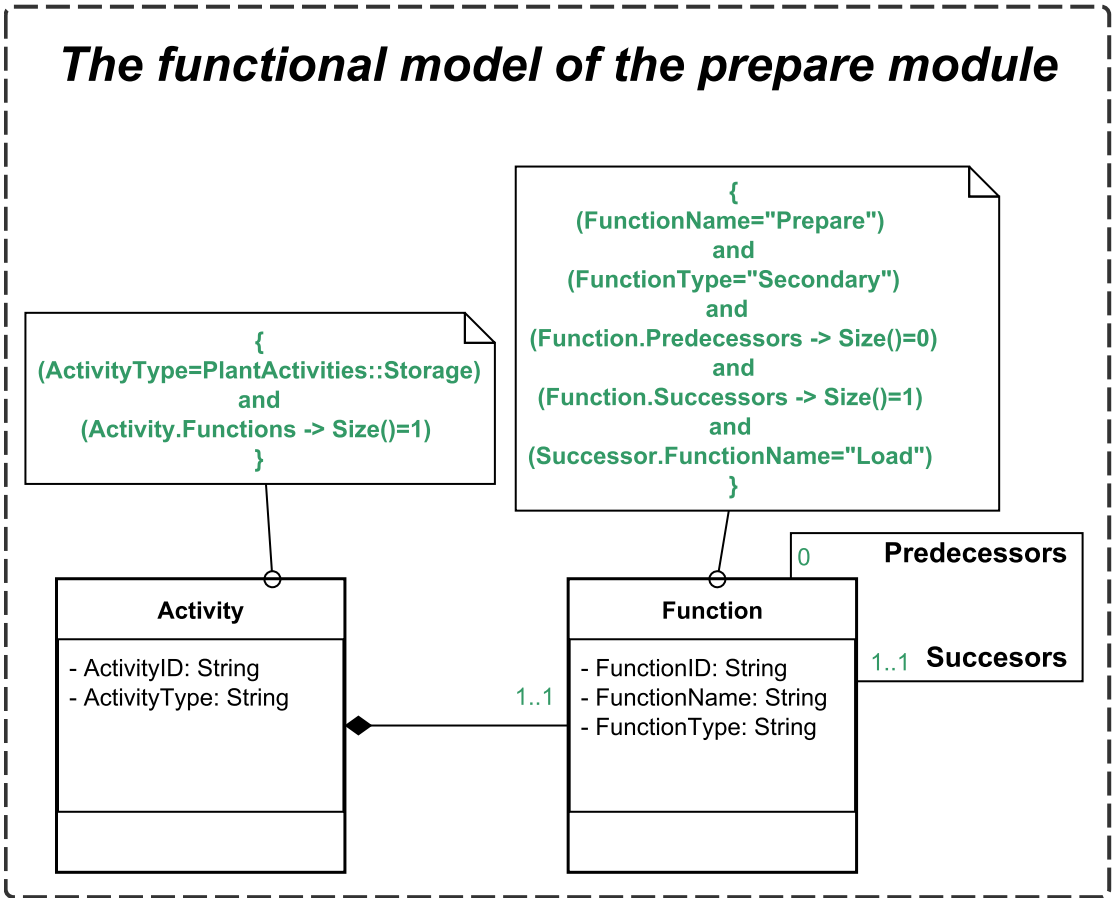

Figure F.1: Model of the function view of the Prepare module.

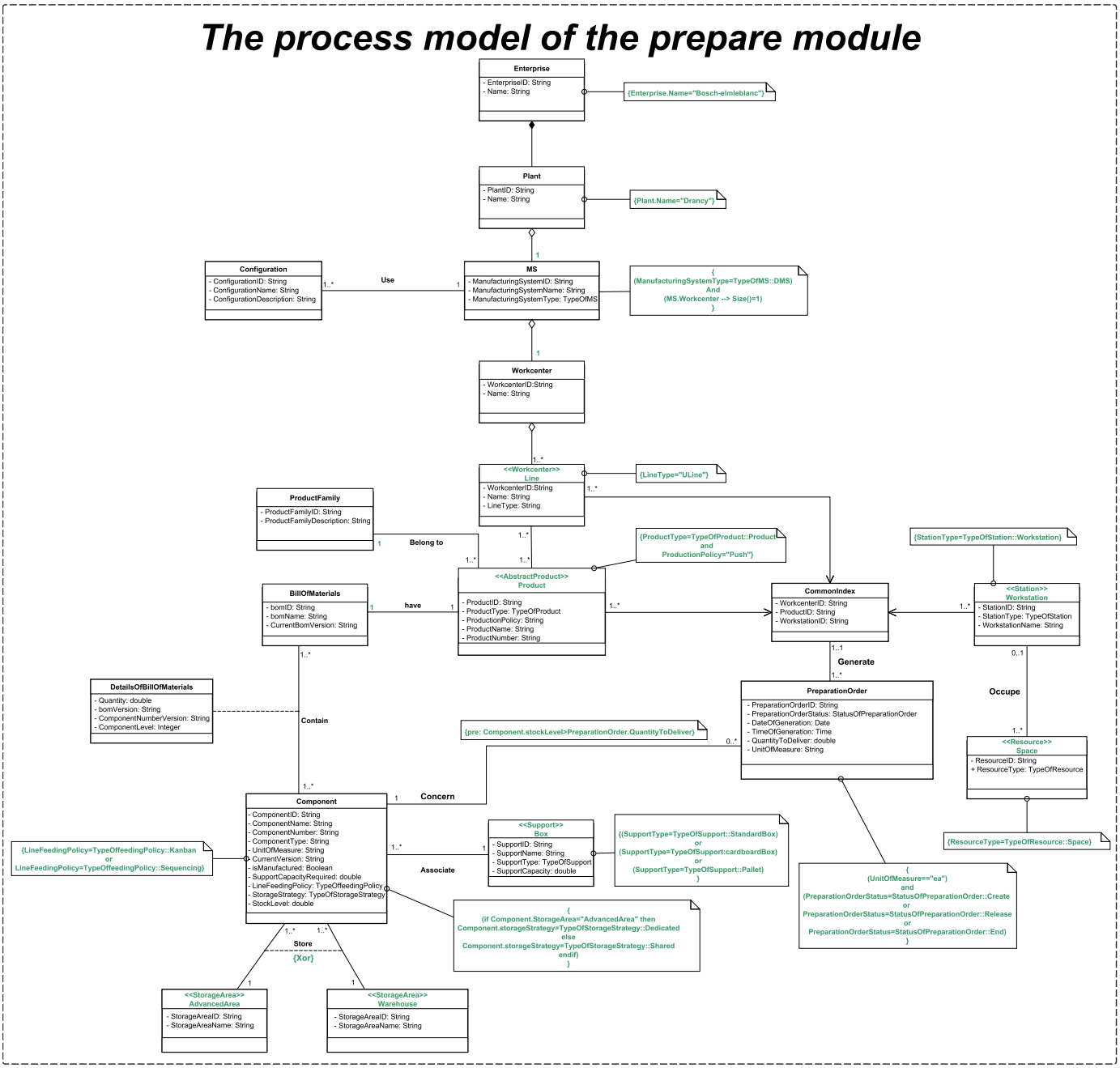

Figure F.2: Model of the process view of the Prepare module.

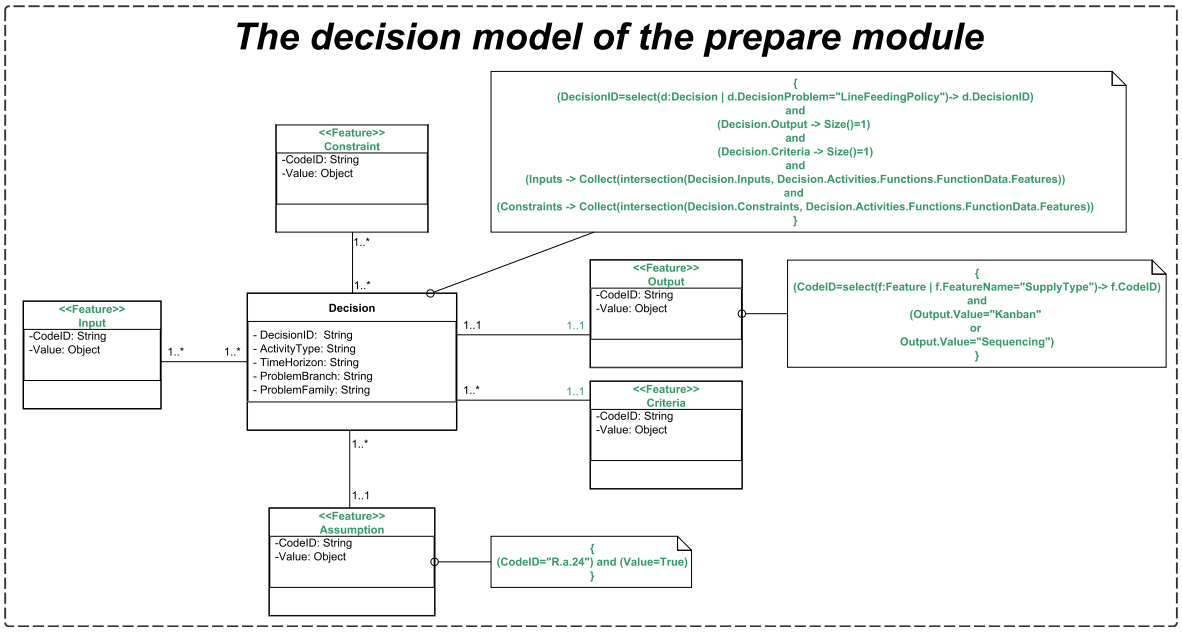

Figure F.3: Model of the decision view of the Prepare module.

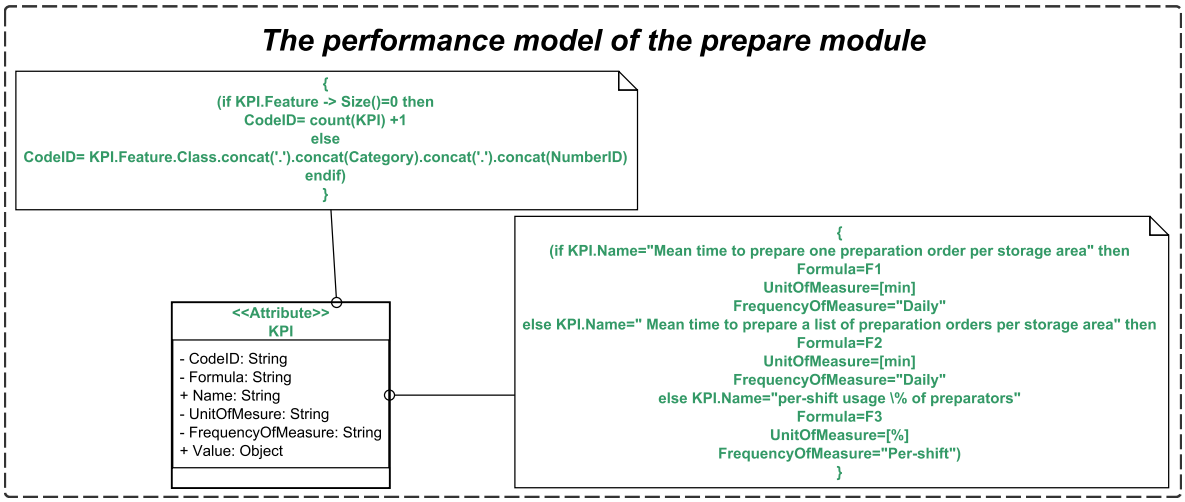

Figure F.4: Model of the performance view of the Prepare module.

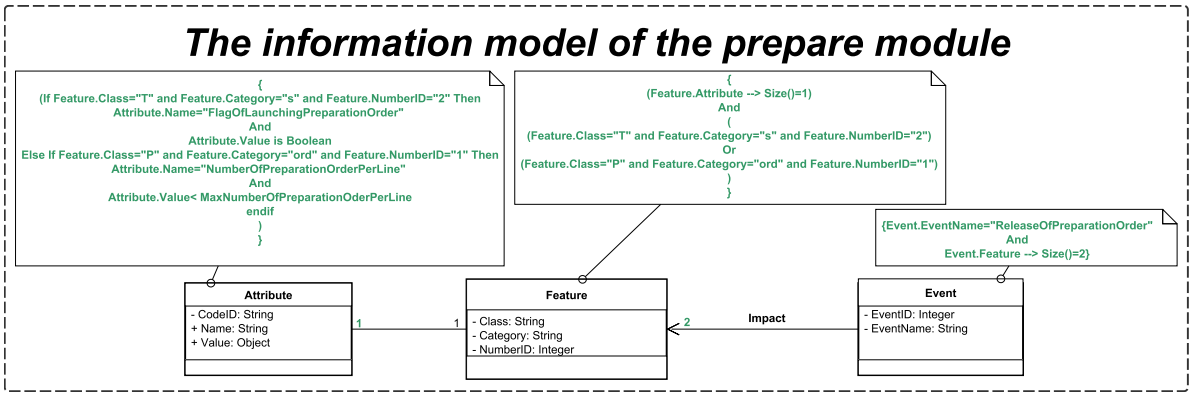

Figure F.5: Model of the information view of the Prepare module.

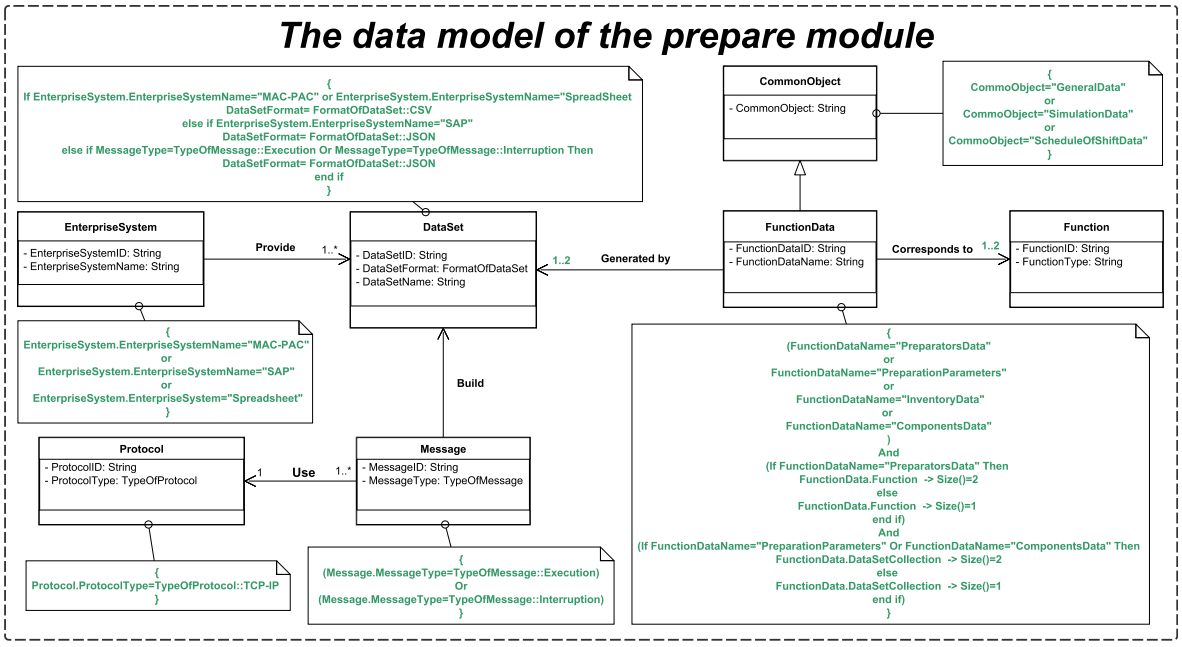

Figure F.6: Model of the data view of the Prepare module.

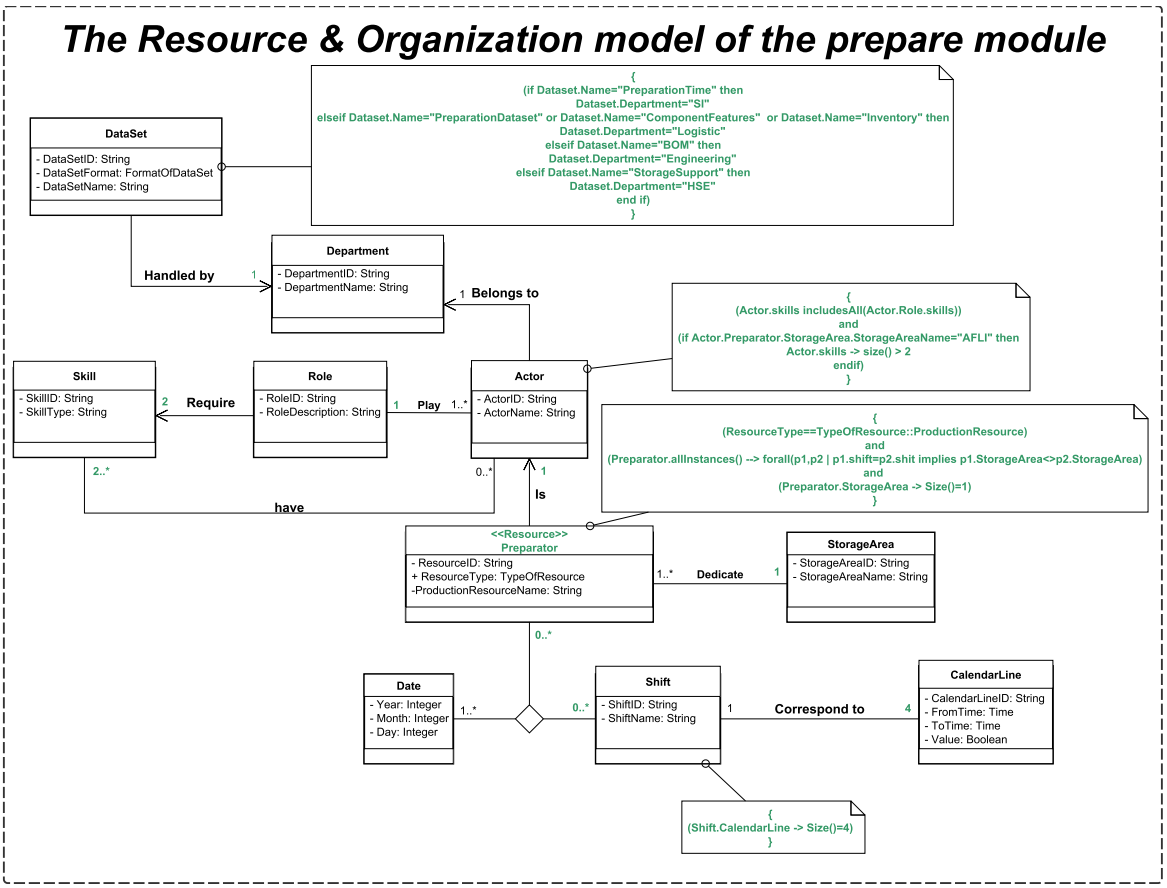

Figure F.7: Model of the resource & organization view of the Prepare module.

F.3 Implementation Phase

F.3.1 Creation of the Data Access Layer

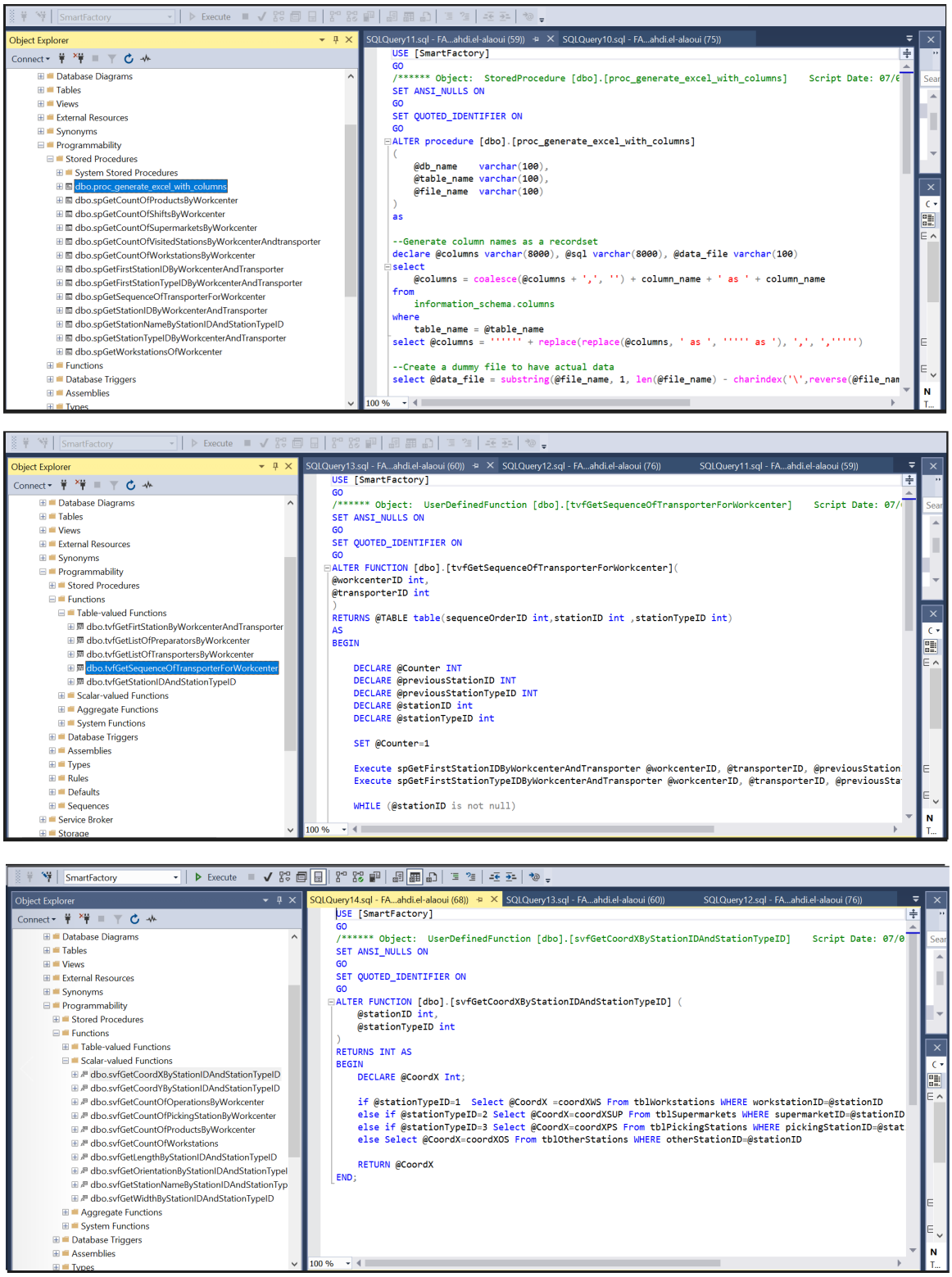

Figure F.8: Screenshot of procedures, table-valued functions, and scale-valued functions defined in database.

F.3.2 List of Algorithms (the Pseudo-Codes)

Algorithm 6 Algorithm of generation of Prepration Order.

procedure GENERATEPREPARATIONORDERSFROMWORSKTATIONS(l, w)
  $n \leftarrow getSizeOfComponentListOfWorkstation(w)$
  for $k \leftarrow 1, n$ **do**
    $LFP \leftarrow getLineFeedingPolicy(C_{lwk})$
    $FlagPRO \leftarrow FlagOfLaunchingPRO(C_{lwk})$
    if $LFP = 1$ && $FlagPRO = 0$ **then** ▷ 1 means KANBAN
      $cap \leftarrow getStorageCapacityOfSupportOfComponent(C_{lwk})$
      if $Inv(C_{lwk}) < cap$ **then**
        $PRO(C_{lwk})$ $Status \leftarrow$ "Create"
        $FlagOfLaunchingPRO(C_{lwk}) \leftarrow 1$
        UPDATETHEPICKINGLIST(l, w, $PRO(C_{lwk})$)
      end if
    else if $LFP = 2$ && $FlagPRO = 0$ **then** ▷ 2 means JIS
      $p \leftarrow getCurrentProduct(l, w)$
      $pp \leftarrow getPlannedProductOfLineAtPeriod(l, T_{cyc} + \delta)$
      $T_{cyc} \leftarrow getCurrentCycle(l, w)$
      $CLS \leftarrow getCurrentLotSizeOfProduction(l, w)$
      $LS \leftarrow getLotSizeOfLine(l)$
      $\delta \leftarrow getOffSetOfLaunchingPRO(l)$
      if $pp = p$ && $LS = CLS$ **then**
        $PRO(C_{lwk})$ $Status \leftarrow$ "Create"
        $FlagOfLaunchingPRO(C_{lwk}) \leftarrow 1$
        UPDATETHEPICKINGLIST(l, w, $PRO(C_{lwk})$)
      end if
    else
      "Line feeding policy not yet supported"
    end if
  end for
end procedure

Algorithm 7 Algorithm of updating the picking list

1: **procedure** UPDATETHEPICKINGLIST(l, w, $PRO(C_{lwk})$)
2: $SA \leftarrow getStorageAreaOfComponent(C_{lwk})$
3: $maxPRO \leftarrow getMaximumNumberOfPreparationOrders(SA, l)$
4: $wTime \leftarrow getWaitingTimeBeforePrintingPRO(SA, l)$
5: $AddPreprationOrderToThePickingList(PRO(C_{lwk}))$
6: $NumberOfPROPerLine(SA, l) \leftarrow NumberOfPROPerLine(SA, l) + 1$
7: $PRO(C_{lwk})$ $Status \leftarrow$ "Schedule"
8: **if** $NumberOfPROPerLine(SA, l) = 1$ **then**
9: $elapsedTimeFromTheFirstPRO \leftarrow T_{now}$
10: **end if**
11: **if** $elapssedTimeFromTheFirstPRO \neq 0$ **then**
12: $Takt \leftarrow getTaktTimeOfLine(l)$
13: $elpasedTimeFromTheFirstPRO \leftarrow elpasedTimeFromTheFirstPRO + Takt$
14: **if** $elapsedTimeFromTheFirstPRO < wTime$ && $NumberOfPROPerLine(SA, l) < maxPRO$ **then**
15: GENERATEPREPARATIONORDERSFROMWORSKTATIONS(l, w)
16: **else**
17: Print $PickingList(i)$
18: $elapsedTimeFromTheFirstPRO \leftarrow 0$
19: $NumberOfPROPerLine(SA, l) \leftarrow 0$
20: **for all** $C_{lwk} \in PickingList(i)$ **do**
21: $PRO(C_{lwk})$ $Status \leftarrow$ "Release"
22: **end for**
RETRIEVECOMPONENTS($PickingList(i), SA$)
23: **end if**
24: **end if**
25: **end procedure**

Algorithm 8 Algorithm of picking components

procedure RETRIEVECOMPONENTS($PickingList(i), SA$)
  $Prep = getIndexsOfPreparatorsAssignedToSA(SA)$
  $NbPreparatorsUnvailable \leftarrow 0$
  for $o \leftarrow 1, Size(Prep)$ **do**
    if $Preparator(o)$ $Status$ = "Available" **then**
      $PREP_{ij} \leftarrow 1$ Allocate Preparator j *to* $PickingList(i)$ $Preparator(o)$ $Status$ = "Unavailable"
      $ExitFor$
    else if
    then
      $NbPreparatorsUnvailable \leftarrow NbPreparatorsUnvailable + 1$
    end if
  end for
  if $NbPreparatorsUnvailable = Size(Prep)$ **then**
   RETRIEVECOMPONENTS($PickingList(i), SA$)
  else if
   then
    $PRO(C_{lwk} \leftarrow getFirstPRO(PickingList(i))$
    while $PickingList(i) \neq \varnothing$ **do**
      $STS \leftarrow getStorageSupportOfComponent(C_{lwk})$
      if $STS =$ "Box" **then**
        Prepare $PRO(C_{lwk})$ in "Box"
      else if $STS =$ "$Pallet$" **then**
        Prepare $PRO(C_{lwk})$ in "Pallet"
      end if
      $PRO(C_{lwk})$ $Status \leftarrow$ "End"
      $PickingList(i) \leftarrow PickingList(i) - PRO(C_{lwk})$
      $PRO(C_{lwk}) \leftarrow getNextPRO(PickingList(i)$
    end while
  end if
end procedure

Algorithm 9 Algorithm of management of picking order

procedure MANAGEPICKINGORDERS(l)
  $NPS \leftarrow getNumberOfPickingStations(l)$
  for $PS \leftarrow 1 to NPS$ **do**
    $NbFullCarts(l, PS) \leftarrow getNumberOfFullCarts(l, PS)$
    for $i \leftarrow 1 to NbFullCarts(l, PS)$ **do**
      $PICO(Cart(i, l, PS))$ $Status \leftarrow$ "Release"
    end for
  end for
  ASSIGNMENTOFMILKRUNDRIVERS(l)
end procedure

Algorithm 10 Algorithm of assignement of milkrun drivers

```
1: procedure AssignmentOfMilkrunDrivers(l)
2:     Trp ← getIndexsOfMilkrunDriverAssignedToLine(l)
3:     for t ← 1, Size(Trp) do
4:         if MilkrunDriver(t) Status = "Available" then
5:             PS ← getFirstPickingStation(t)
6:             capTrp(l) ← getCapacityOfMilkrunDriver(t, l)
7:             NbCartsLoaded(t) ← 0
8:             CartListToUnload(t) ← ∅
9:             while NbCartsUploaded(t) < capTrp(l) && NbFullCarts(l, PS) <> 0
   do
10:                MoveTo PS
11:                MilkrunDriver(t) Status ← "Unavailable"
12:                Tow Cart(i, l, PS)
13:                PICO(Cart(i, l, PS)) Status ← "End"
14:                CartListToUnload(t) ← CartListToUnload(t) + Cart(i, l, PS)
15:                NbCartsLoaded(t) ← NbCartsLoaded(t) + 1
16:                NbFullCarts(l, PS) ← NbFullCarts(l, PS) − 1
17:                if NbFullCarts(l, PS) = 0 then
18:                    NextPS(t) ← getNextPickingStation(t, l)
19:                    if NextPS(t) ≠ ∅ then
20:                        PS ← NextPS(t)
21:                        MoveTo PS
22:                        NbFullCarts(l, PS) ← getNumberOfFullCarts(l, PS)
23:                    else
24:                        SUP ← getFirstSupermarket(t, l)
25:                        while SUP ≠ ∅ do
26:                            MoveTo SUP
27:                            z ← getIndexOfSupermarket(SUP)
28:                            ComponentList ← getComponentListToUnload(CartListToUnload(t), z)
29:                            w ← getIndexOfWorkstationAssociatedToSupermarket(z)
30:                            for k ← 1toSize(ComponentList) do
31:                                STS ← getStorageSupportOfComponent(C(lwk))
32:                                Unload STS
33:                                capStorage ← getStorageCapacityOfSTS(C(lwk))
34:                                Inv(C(lwk)) ← Inv(C(lwk)) + capStorage
35:                                if STS = "Box" then
36:                                    Add STS to EmptyBoxes
37:                                else if STS = "Pallet" then
38:                                    Add STS to EmptyPallets
39:                                end if
40:                            end for
41:                            SUP ← getNextSupermarket(t, l)
42:                        end while
43:                        MilkrunDriver(t) Status ← "Available"
44:                        ManagePickingOrders(l)
45:                    end if
46:                end if
47:            end while
48:        end if
49:    end for
50: end procedure
```

F.3.3 Building a Simulation Model of Modules: Results

Table F.18: Results of the goodness of fit using the Kolmogorov-Smirnov test for the variables X1,X3, and X5.

X1			X3			X5		
Distribution	**statistic**	**pvalue**	**Distribution**	**statistic**	**pvalue**	**Distribution**	**statistic**	**pvalue**
triang	0.01953	0.63395	halfcauchy	0.19627	2.99E-09	uniform	0.02041	0.57792
powernorm	0.02188	0.48789	pareto	0.20502	4.63E-10	gennorm	0.02157	0.50680
gengamma	0.02202	0.48001	alpha	0.21462	5.44E-11	powerlaw	0.02462	0.34082
gamma	0.02222	0.46798	powerlognorm	0.22504	4.73E-12	gausshyper	0.03063	0.13078
erlang	0.02223	0.46768	genextreme	0.22774	2.47E-12	beta	0.03179	0.10605
norm	0.02249	0.45242	expon	0.23548	3.64E-13	pearson3	0.06722	4.14E-06
exponnorm	0.02249	0.45241	halfgennorm	0.24121	8.44E-14	logistic	0.06782	3.28E-06
t	0.02250	0.45215	genexpon	0.24686	1.93E-14	loggamma	0.06839	2.62E-06
loggamma	0.02264	0.44412	chi2	0.26795	5.66E-17	t	0.06869	2.32E-06
pearson3	0.02298	0.42502	pearson3	0.26795	5.66E-17	powerlognorm	0.06891	2.13E-06
lognorm	0.02324	0.41092	gamma	0.26795	5.66E-17	exponnorm	0.06893	2.11E-06
rdist	0.02773	0.21354	erlang	0.26795	5.66E-17	norm	0.06893	2.11E-06
beta	0.03004	0.14521	gennorm	0.26853	4.79E-17	gamma	0.06906	2.00E-06
gausshyper	0.03073	0.12857	gausshyper	0.27753	3.38E-18	erlang	0.06908	1.99E-06
gennorm	0.03076	0.12795	logistic	0.28869	1.11E-19	alpha	0.07137	7.85E-07
logistic	0.03147	0.11248	loglaplace	0.29751	6.75E-21	genextreme	0.07299	3.99E-07
alpha	0.03705	0.03720	exponnorm	0.30101	2.16E-21	chi	0.09584	5.59E-12
laplace	0.05279	0.00062	cauchy	0.30740	2.61E-22	rdist	0.10084	3.23E-13
exponweib	0.05605	0.00022	halflogistic	0.30969	1.21E-22	loglaplace	0.10828	3.54E-15
loglaplace	0.06687	4.74E-06	chi	0.31290	4.07E-23	laplace	0.11039	9.24E-16
cauchy	0.08802	3.60E-10	laplace	0.33721	7.11E-27	truncexpon	0.11380	1.01E-16
genhalflogistic	0.12012	1.38E-18	loggamma	0.33947	3.07E-27	cauchy	0.12897	2.26E-21
uniform	0.12029	1.22E-18	t	0.34009	2.43E-27	halflogistic	0.13206	2.18E-22
powerlaw	0.16656	1.97E-35	norm	0.34010	2.43E-27	triang	0.14812	4.45E-28
truncexpon	0.20208	4.25E-52	halfnorm	0.34404	5.49E-28	genexpon	0.16503	8.62E-35
halflogistic	0.23334	1.58E-69	exponpow	0.35110	3.68E-29	expon	0.17175	1.16E-37
halfnorm	0.23536	9.54E-71	truncexpon	0.35756	2.92E-30	pareto	0.18219	2.33E-42
genexpon	0.26518	5.35E-90	rdist	0.36171	5.59E-31	genhalflogistic	0.18322	7.76E-43
expon	0.26892	1.39E-92	powerlaw	0.42863	6.82E-44	halfnorm	0.22988	1.80E-67
halfcauchy	0.28272	1.82E-102	uniform	0.43667	1.28E-45	halfcauchy	0.25957	3.44E-86
pareto	0.34636	4.74E-155	triang	0.44568	1.34E-47	halfgennorm	0.26131	2.32E-87
halfgennorm	0.35274	5.69E-161	exponweib	0.51017	2.76E-63	gengamma	0.44698	3.00E-263
powerlognorm	0.39054	1.30E-198	genhalflogistic	0.51457	1.87E-64	chi2	0.91679	0
chi	0.99548	0	beta	0.61752	5.20E-96	exponweib	0.57462	0
chi2	0.99359	0	lognorm	0.64445	9.46E-106	exponpow	0.81249	0
exponpow	0.89576	0	gengamma	0.67340	4.94E-117	lognorm	0.52523	0
genextreme	0.81230	0	weibull_max	0.69279	4.17E-125	powernorm	0.89459	0
truncnorm	1	0	weibull_min	0.76197	5.23E-158	truncnorm	1	0
weibull_min	0.78051	0	powernorm	1	0	weibull_min	0.68442	0
weibull_max	0.78448	0	truncnorm	1	0	weibull_max	0.68549	0

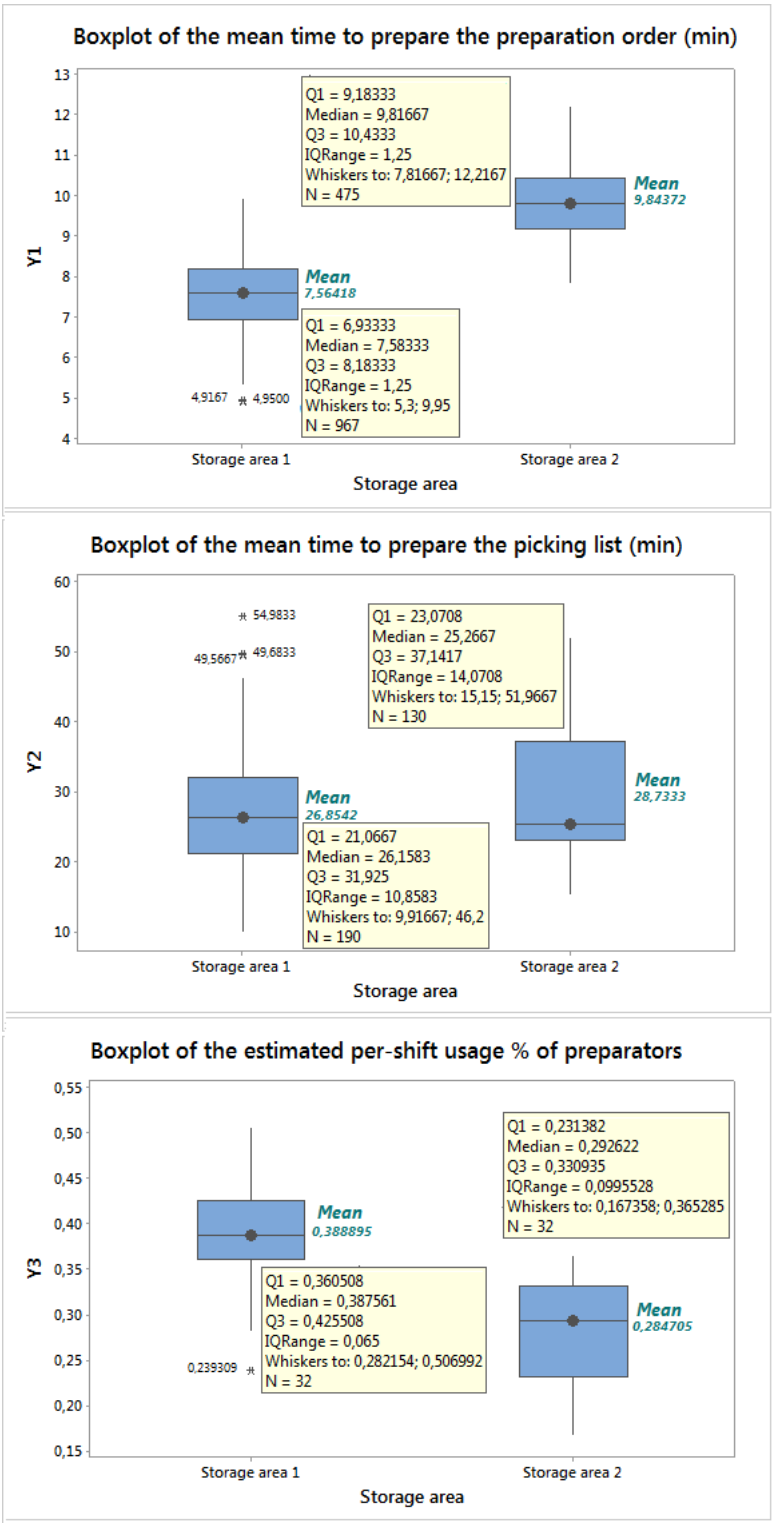

Figure F.9: The box plot of the output variable Y1,Y2, and Y3 of the Prepare module.

F.3.4 Sensitivity Analysis for Variables: X1 and X2

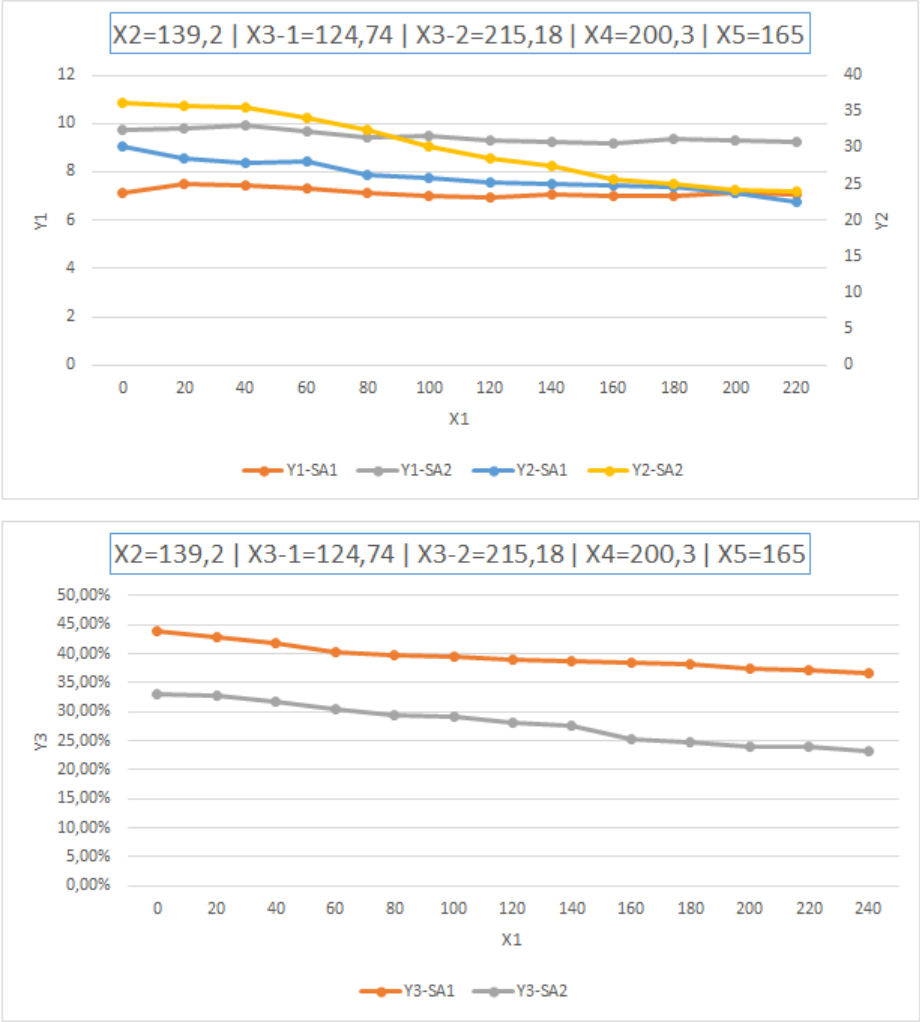

Figure F.10: Sensitivity analysis related to the X1 variable.

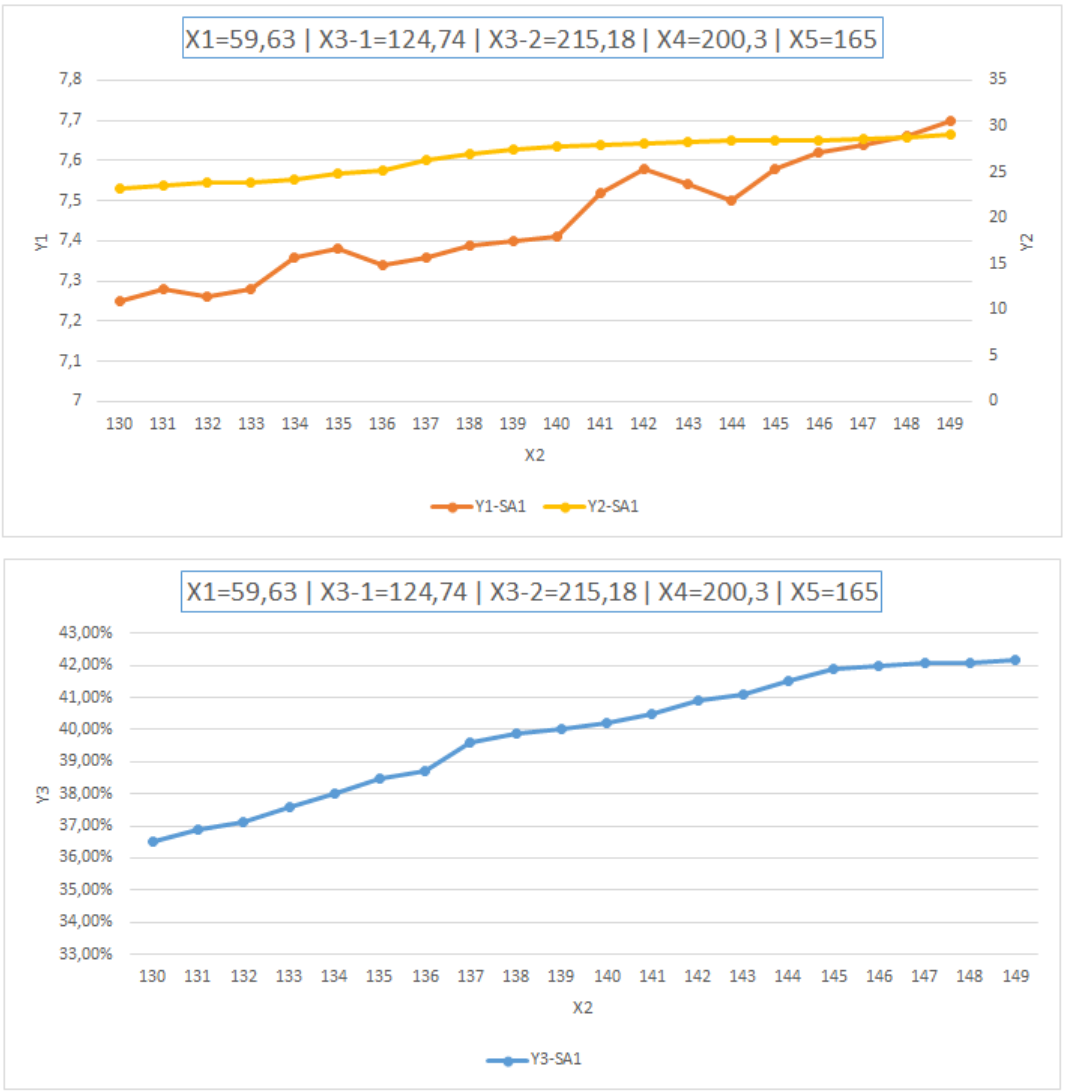

Figure F.11: Sensitivity analysis related to the X2 variable.

F.4 Maintenance Phase

F.4.1 Documentation and Versioning of Module Models

- **Workcenter/line:**

  The workcenter and lines appear as an attribute of several entities.

Table F.19: Attributes of the Workcenter/Line element.

Entities
Preparation order
Inventory
Box check
Control shift & resources
Picking order
Control picking order
Empty Box
Control Transporters
Manufacturing order
Control WIP
Product

- **Workstation:**

  A workstation is modeled as a "*Station*". It has all attributes of *Station building block*. The expressions "*exLotSize (aCommonIndex)*" and "*exProcessTimeOfProduct (aCommonIndex)*" return the lot size and the processing time of the product, respectively. Thanks to aCommonIndex, we can track the product in production in a specific workstation, which belongs to a specific line of the workcenter.

- **Product:**

  A product is modeled as an entity with the attributes defined in Table F.20.

- **Component:**

  A component is modeled as an entity with the attributes defined in Table F.21.

- **BOM:**

  BOM is modeled as an advanced set "asListAllComponentsOfLines" with a tree data structure. The number of rows of this advanced set is equal to the number of lines

Table F.20: Attributes of the product element.

Attributes	Definition
aCommonIndex	a unique ID corresponding to the "Workcenter/Line, Workstation and Product"
aProductionPolicy	1:Push 2:Pull
aPicture	Picture of the product

Table F.21: Attributes of the component element.

Attributes	Definition
aLabelOfComponent	The unique reference of a component
aSupermarket	The supermarket where the component will be delivered
aCapSupOfSupportOfComponent	The storage capacity of supermarket in terms of the component storage support
aStorageCapacity	The capacity of the storage support for each component
aStorageSupport	The type of storage support of the component
aSupplyType	The supply policy associated with each component 1:"Kanban" 2:"JIS"
aUnitQuantity	The quantity of the component needed to assemble one product.
aTriggeringQuantity	Quantity level that trigger the generation of preparation order.
aDeliveredQuantity	The quantity to deliver to the workstation when the preparation order is launched.
aPicture	Picture of the component

of the workcenter. Each row references another advanced set, which contains a list of expressions. The rows of each expression contain a list of the component in a specific workstation. Columns of the expression allow us to distinguish different products managed in the workstation.

- **Resource (Preparator):**

  The Preparator is modeled as an active resource with all attributes of resource building blocks. The preparator works on a list of preparation orders (batch), and the processing is done per unit of preparation order. Therefore, the preparation order (entity) seizes the preparator (resource) assigned to the storage area receiving the preparation order. A set is created for each storage area to specify the preparator list assigned to it.

- **Storage area:**

  The storage area appears as an attribute of the preparation order and component entities.

- **Storage support:**

  Storage support is modeled as an entity with the attributes defined in Table F.22.

Table F.22: Attributes of the storage support element.

Attributes	Definition
aType	1:"Standard box" 2:"Cardboard box" 3:"Pallet"
aWidthOfSupport	Width of the storage support
aLengthOfSupport	Length of the storage support
aHeightOfSupport	Height of the storage support

- **Supermarket:**

  It is modeled as a "*Station*". Therefore, it has all attributes of the Station building block.

- **Process:**

  The preparation process is modeled using a Delay building block. A constant unit time or distribution law is allocated to the processing time, which depends on two factors, namely "vNbOfSupport" and "aStoratgeType" of the component. The expression "*exExternProcessTime (Process, aStoratgeType)*" allows retrieving the unit times associated with each type of storage support.

- **Date:**

  The date is modeled as an entity with the attributes defined in Table F.23.

  Table F.23: Attributes of the date element.

Attributes	Definition
aYear	The year of date
aMonth	The month of date
aDay	The day of the date

  VCurrentDate is a variable that allows having the current date at any time of the simulation. It is incremented by one after each number of hours specified per day for the simulation. We use the variable vNumberOfHoursPerDay to store the number of hours per day.

- **Shift:**

  The shift is modeled as an entity. The number of entities to create is based on the maximum number of shifts per day used during the simulation. Table F.24 defines the attributes of shift.

  Table F.24: Attributes of the shift element.

Attributes	Definition
aAvailableTime	The opening time of each shift per day
aBreakTime	The break time of each shift in the day

  vCurrentShift is a variable that provides the current shift. It is initialized when vCurrentDay is incremented.

- **Schedule:**

  The Schedule is modeled as an advanced set "*asScheduleOfShift*" with the number of rows equal to the number of shifts. In addition, each item (row) has the attributes defined in Table F.25.

  Table F.25: Attributes of the schedule element.

Attributes	Definition
Duration	Time expressed in minutes
Value	It is a boolean used to indicate whether the shift is available for working or not during the duration time.

F.5 Operational Stage: What-if Analysis for the Increase of Customer Demand

In this second use case, we are interested in performing the what-if analysis to increase customer demand for products assembled in line "L4" of the studied elm.leblanc plant. Therefore, the top management needs to verify whether the current configuration of the study assembly line can offer an appropriate response.

F.5.1 Definition Phase

First, the management request is analyzed. In this use case, elm.leblanc is interested in assessing the line "L4" performance when the customer demand increases. Note that increasing the line capacity impacts automatically the internal logistic, and the managers seek to identify how.

When dealing with this use case, three modules already existed in the library modules (see Figure 5.1). Regarding our analysis, we judged that the three modules are required for the use case at the study. First, we checked their version and ensured that are updated. Then, we identified that a new module must be developed, namely, the produce module, since the simulation model must include the entire MS process: preparation, loading, transport, and production activities.

So, we go back to the strategical/tactical stage for developing the new produce module according to the roadmap outlined in Chapter 4. This development has been anticipated in Section 5.2 for simplification reasons as explained in the Section 5.1. Therefore, no further changes are required for the produce module.

After checking whether the versions of the modules are up to date or not, we could now build the integrated model in the design phase.

F.5.2 Design Phase

The assembly line settings are derived together with elm.leblanc (managers, experts, operators) to comply with elm.leblanc guidelines (e.g., The considered assembly line produces two main products according to a push policy between workstations).

After that, the integrated model involving the four modules (prepare, load, transport, and produce) is assembled and instantiated to derive the case-specific simulation model.

We follow the same approach used in the first use case, i.e., we instantiated the model of each view. The integration of these instances is ensured by the compatibility of views approved in the Chapter 4.

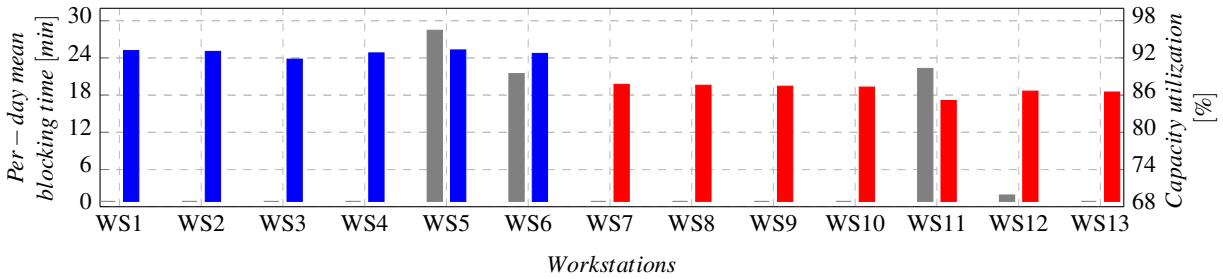

Figure F.12: Per-day mean blocking time (grey) and capacity utilization (blue, red if critical) of workstations.

F.5.3 Implementation Phase

This phase aims to run the tests, validate the model results and discuss them. In this use case, a what-if analysis is performed to assess how the plant reacts when the demand increases w.r.t. the production planning.

The tuning of the instance model is done by trying to reproduce the daily operations of the MS process.

The production of one week (5 days) is simulated; Ten replications are run. Throughout the simulated week, achieved production varies between 89.5% and 96.8% of the targeted demand.

The KPIs defined for the production activity (Capacity utilization and per-day mean blocking time of each workstation) are inspected to find the root causes of this production gap (Figure F.12).

Three workstations have a high blocking time due to a lack of components, causing the capacity utilization of all workstations from WS7 on to decrease considerably.

Traceability files then allow tracking components' inventory levels and detecting the critical ones (Table F.22).

Table F.26: Critical components of workstations with non-zero mean blocking time.

		Characteristics			
Workstations	**Components**	**Box capacity**	**Storage capacity**	**Triggering qty**	**Delivered qty**
WS5	W	12	3	18	12
WS6	X	8	4	24	8
WS11	Y	12	2	24	12
WS12	Z	12	4	24	12

F.5.4 Maintenance Phase

Some solutions are recommended based on the results of the implementation phase.

As for capacity, we could increase either the component box capacity or storage capacity; as for milkrun deliveries, we could increase either the triggering quantity or the delivered quantity.

Since both would heavily impact the process, they can be simulated by a new instantiation of the integrated model before being deployed in the plant. The flexibility and modularity of the integrated model can be fully exploited, and more configurations can be tested with the different tuning of the parameters.

Finally, the model documentation and the versioning are updated.

F.6 Operational Stage: Coupling the Simulation and Optimization for Addressing the ALBP with KPI 4.0 (Ergonomics)

This last use case aims to provide elm.leblanc with a robust balancing solution for the line design of the new product generation called "DNA". We use a simulation approach as a postponement strategy to check whether the assembly line balancing solution proposed by the optimization model would allow for meeting the customer demand. The aim is to stress the manual task times to reflect reality and observe the system behavior before implementing the solution.

F.6.1 Definition Phase

First, we analyzed the availability of the modules in the library of modules. As the objective of this use case is to focus only on line balancing, we decided to include only the produce module. However, management has requested that ergonomics be included as a KPI to evaluate the performance of the assembly line in addition to productivity.

Therefore, we need to update the produce module to include the new KPI and be able to compute it in the traceability file.

On the other hand, we need to develop an optimization model to solve the ALBP problem by considering the ergonomics criteria during the design phase.

Hereafter, the Appendix F.6.2 details the produce module's update and the optimization model's development.

F.6.2 Back to the Development Cycle

Remember that only the steps that are impacted must be reviewed. Therefore, we will focus on steps involving the performance and decision views (C.f. Figure E.1).

F.6.2.1 Definition Phase of Strategical/Tactical Stage

In addition to classical KPIs defined in Table 5.6, we added new KPIs related to the produce module to evaluate the objective function during the simulation.

- **Define new KPIs of the Produce module:**

  To assess the ALBP decisions, we consider the classical KPIs and the specific ones related to ergonomics. We consider a KPI assessing the evolution of the ergonomics level, defined with Equation F.1 after several takt time to evaluate the evolution during a shift of production and determine whether or not the operator is at risk of developing musculoskeletal disorders and risks of injuries.

  Besides, we compute the utilization rate of resources since this indicator reflects the repartition of the percentage of working time in each workstation and allows for detecting bottlenecks.

  Also, we compute the filling rate to reflect the percentage of customer demand satisfaction, which allows for judging the flexibility of assembly lines.

F.6.2.2 Design Phase of Strategical/Tactical Stage

- **Build the KPI structure and specify the data required to compute them:**

  The utilization rate of each resource in the workstation is computed as follows:

  In one takt time, a fraction between the total time of tasks assigned to the workstation and takt time.
  In one day, by total occupied time of resource in each workstation and availability time of this resource.

  The filling rate will be computed by a fraction of the total quantity produced and the quantity asked by customers.

  During the simulation, the ergonomics KPI is defined with a measure representing the evolution of Equation F.1. The idea is to check if the level of ergonomics falls below a risk threshold, denoted F_{risk} and defined by the ergonomist of the company. A given worker in workstation k is at risk if the level of muscular capacity falls below the threshold of risk, i.e., $F_k \leq F_{risk}$ in a given work cycle. A worker at risk could not complete operations correctly and may develop risks. Besides, a worker at risk would increase the production's error rate and quality defects.

- **Define the representation of KPIs:**

  To represent the utilization rate of each resource, we use the bar chart that directly compares solutions of SALBP and the model based on ergonomy criterion.

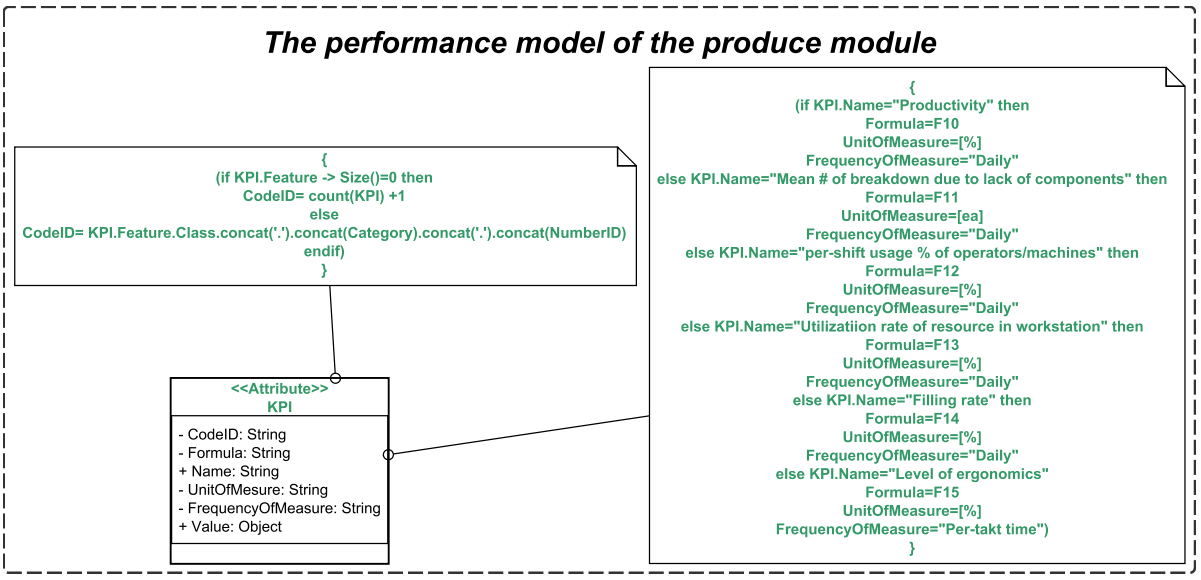

Figure F.13: Update of the performance model of the Produce module.

On the other hand, we use tables to represent the filling rate and the percentage of workstations at risk.

- **Instantiation of the meta-models of views:**

  We updated the performance model of the produce module as shown in Figure F.13 by adding the three KPIs defined before.

  Likewise, Figure F.14 illustrates the update of the decision model of the produce module. This model adds the criteria R.o.3 consisting of minimizing the risk of fatigue and the new assumption R.a.5 of the level of ergonomy that we set to 15%. Note that the operation workloads evaluated by the borg scale are captured as the inputs via the OCL expression, indicating the intersection between the inputs of the line balancing decision and the features considered in the function data of the produce module.

F.6.2.3 Implementation Phase of Strategical/Tactical Stage

- **Build an optimization model:**

  The industrial assembly line manufactures a single condensing boiler, assembled in a serial straight line, with 300 tasks. Although the process planning department provides input data for this use case, the input data (description of tasks, processing times, precedence diagram, takt time, and tasks' load) will not be disclosed for confidentiality reasons. The physical load of tasks is obtained with the Borg scale (Borg, 1982). A qualified ergonomist from the company evaluates the load of the

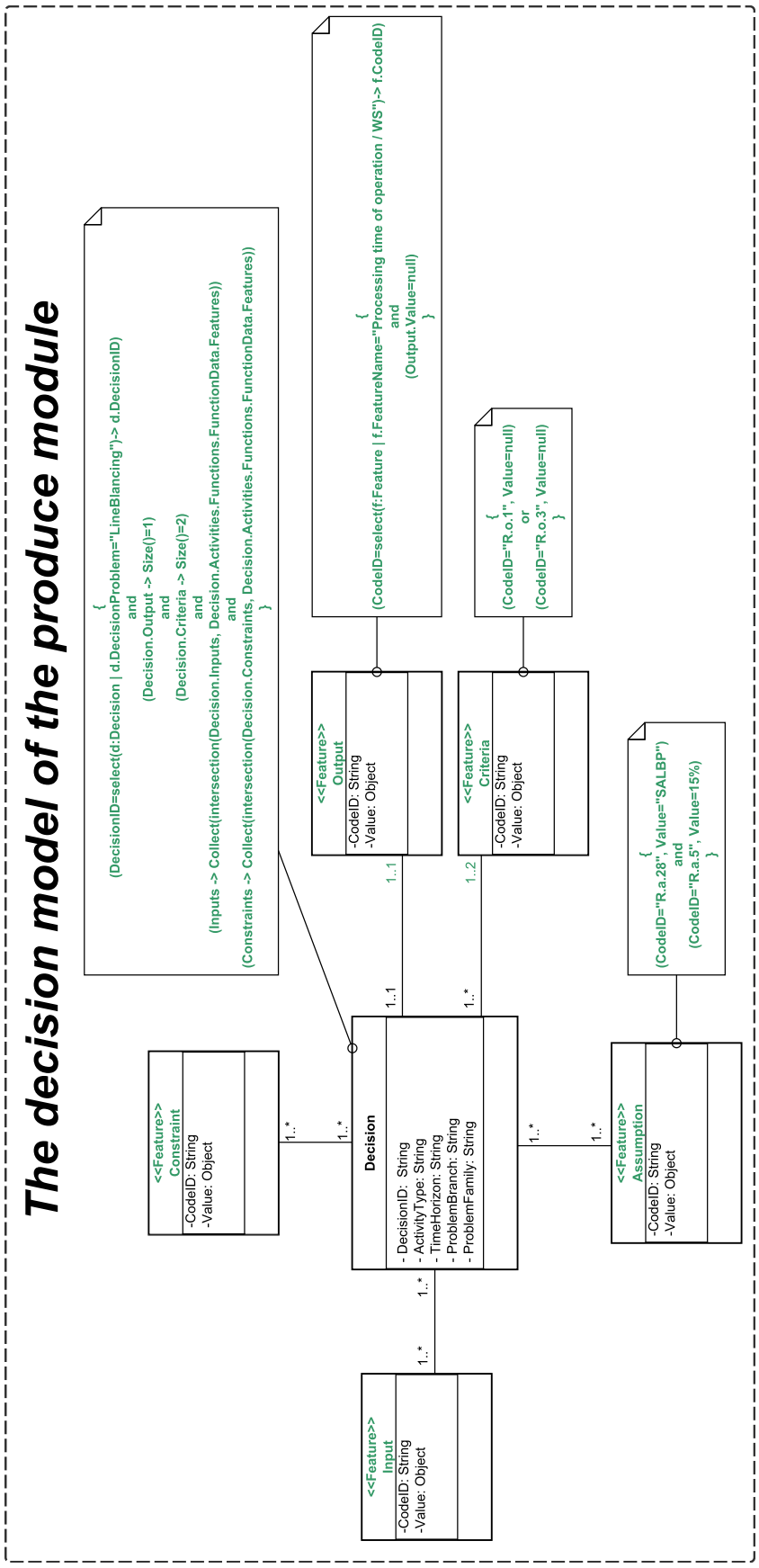

Figure F.14: Update of the decision model of the Produce module.

tasks with the active involvement of the process planning department and health, safety, and environment department.

The decisions are the assignment of tasks to workstations, and the constraints are the respect of the customer's demand, represented by the takt time, and the respect of the technological precedence constraints between tasks. Finally, the objective is the optimization of ergonomics.

We model the case study as a SALBP (cf., Baybars (1986) for more details on the problem hypothesis and classical linear formulation). We use the fatigue and recovery model developed by Ma et al. (2010) as a criterion for assessing the ergonomics level in workstation k, with $k \in W$ and W the set of workstations. The ergonomics criterion (i.e., the fatigue and recovery criterion) after one takt time in a given workstation is represented in the following equation:

$$F_k = 1 + (\mathrm{e}^{-\alpha} - 1)\mathrm{e}^{-\beta} \quad \forall k \in W \tag{F.1}$$

With α the workload assigned to the workstation k, and β the length of recovery time after the completion of the working time (i.e., the idle time). $F_k \in [0, 1]$ represents the level of ergonomics in workstation k after one takt time, depending on the load of operations assigned to the corresponding workstation, modeled with α. We define the objective function to maximize the ergonomics level at the most charged workstation (i.e., workstation k when the worker presents the lowest level of ergonomics: $Min_{k \in W} \{F_k\}$, we refer to the work of Abdous (2019); Abdous et al. (2018a,b) for more details on the ergonomics model, details about the formulation of this objective function and different hypothesis.

- **Choose the solving approach:**

  We used metaheuristics to solve the problem. Indeed, the SALBP is an NP-Hard combinatorial problem. In addition to the combinatorial aspects of the problem, the objective function is non-linear. Metaheuristics come in handy for such combinatorial problems with the industrial size. Of course, the metaheuristic does not guarantee optimality but could achieve acceptable results in a reasonable computational time.

  We propose a metaheuristic based on the well-known framework of Iterative Local Search (ILS) (Stützle and Ruiz, 2017). ILS is a multi-start-based metaheuristic that iterates a specific local search procedure from different starting solutions to sample various regions of the search space and avoid local optimum. Local search explores the search space, moving from a solution to neighboring solutions, and seeking to improve the objective function. We apply the classical swap sequentially and shift

the neighborhood for operations, as defined in Scholl (1999). We refer to Abdous (2019) for extensive details on the ILS algorithm.

- **Conduct numerical experiments:**

  We perform a series of numerical experiments with instances from the literature to validate the behavior.

  To assess the performance of the ILS algorithm, we compare the result obtained with the metaheuristic ILS with the consideration of ergonomics and the result obtained with SALBP without any ergonomics measure (solved with Cplex). We compare two values, the percentage of working time (i.e., the proportion of processing time in a given workstation) and the percentage of workload (i.e., the proportion of workload assigned to a workstation). The balancing solution presents 14 workstations in this case study. The distribution of the percentage of working time is presented in Figure F.15, and the percentage of workload in Figure F.16. These results are obtained from the optimization model after one takt time.

  The distribution of working time demonstrates that the working time of the solution obtained with ILS is more balanced and does not exceed the theoretical average time (represented by the yellow line), which suggests that the solution of ILS offers more balanced recovery time for workers and is potentially more robust against hazards during the various production cycles. On the other hand, the SALBP working time distribution presents more dispersion than the average value, with nine workstations exceeding the average working time values. Concerning workload distribution, although we observe more dispersion in both solutions, the ILS's solution seems more equitable in the assignment of workload than the SALBP solution.

- **Build the simulation model:**

  We update the produce module by adding the new KPIs to be computed. Then, we add some attributes (level of risk, task time) in the traceability file generated by the produce module. As the production process does not change, we consider the last version of the produce module as valid. Therefore, we do not need to launch experimentations or corner tests.

- **Map decision variables of the optimization model with the simulation model attributes:**

  We solved the optimization model that provides the values X_{jk}, which specify whether task j is assigned or not to the workstation k. This decision variable will allow computing the total time within each workstation.

  Table F.27 shows the mapping between the decision variable X_{jk} and the attributes "*exProcessTimeOfProduct (aCommonIndex)*" (see the documentation of

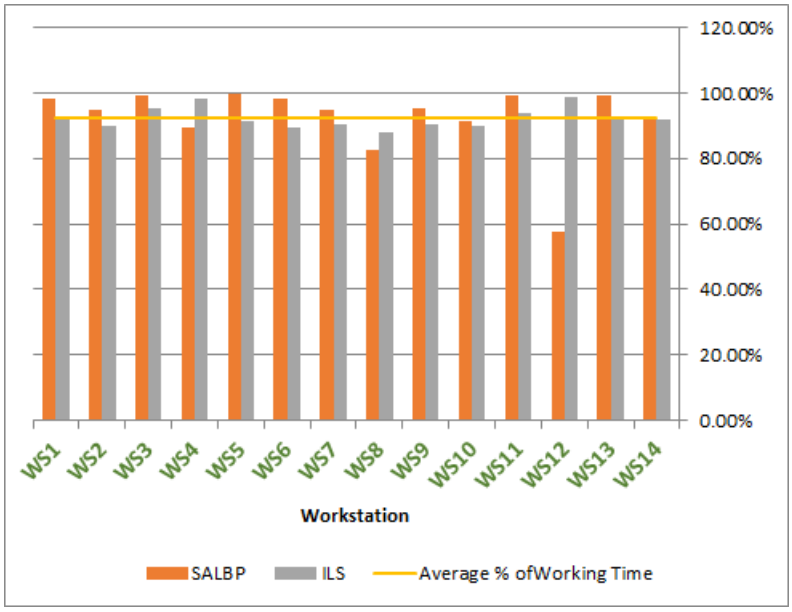

Figure F.15: % of working time in different workstations for SALBP and ILS solutions.

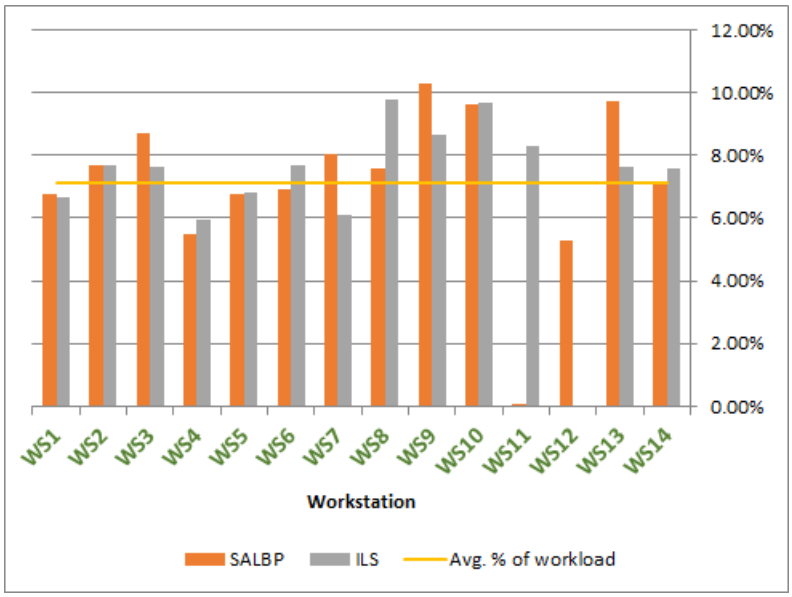

Figure F.16: % of workload in different workstations for SALBP and ILS solutions.

the workstation element in Subsection 5.2.4.1). Remember that aCommonIndex is an aggregation ID of Workcenter/Line ID, Workstation ID, and product ID. In our use case, we are developing a new line, "L5", and we have only one product, "1". Therefore, the *aCommonIndex* corresponding to the workstation k will be: "5" & k & "1".

- **Display KPIs and results:**

  Based on the KPIs defined above, we develop a statistic module to generate the dashboard of results obtained by the optimization model. This statistic module uses the data model defined in the data view to retrieve the input data and use the solution files provided by the metaheuristic to obtain the output data. First, the simulation results are generated in different CSV files, and then each is processed to extract the KPIs.

F.6.2.4 Maintenance Phase of Strategical/Tactical Stage

- **Documentation & versionning:**

  We use git as a platform to manage the versions of models. Once we obtain feedback from elm.leblanc managers and experts about the balancing solution implemented, we will adjust the model if needed.

- **Review meetings with team project:**

  Based on the analysis of KPIs during review meetings, the team decided to validate the version of the simulation model. One of the future objectives is to introduce new KPIs, such as line efficiency, computed according to Bosch Production System (BPS) guidelines.

F.6.3 Design phase

We define the assembly line to produce one family of products according to a push policy between workstations. The assembly line layout is assimilated into a straight line, and the production planning is organized according to two daily shifts.

In addition to classical KPIs, we add the new KPIs to the produce module to evaluate the objective function during the simulation.

Afterward, we instantiate the updated models of each view of the produce module using these settings.

F.6.4 Implementation phase

We use simulation software to generate the simulation model of the use case. This model is instantiated from the new version of the produce module according to the assembly line characteristics and configured based on the result of optimization.

Table F.27: Mapping decision variables of the optimization model with attributes of the simulation model of the Produce module.

Module	Variable	Data Set Collection	Function Data	MS Feature Code	Corresponding attributes in the simulation model	Mapping with modeling choices
Produce	$\sum_{j=1}^{n} X_{jk} \quad \forall k \in W$	AssemblyDataSet	AssemblyTimes	T.p.2	exProcessTimeOfProduct ("5" & k & "1")	Processing time of the product "1" in the workstation k of the line "5".

Afterward, a what-if analysis was performed to take into account the uncertainty in the processing time of tasks. First, we use deterministic task times to check the model's behavior. Then, to test its robustness, we stress it using stochastic task times.

The time horizon of the simulation is set to one to compare simulation results with the daily demand. Finally, the simulation is run with 100 replications and a warm-up period of 40 minutes.

Two simulation model instances are generated to represent the SALBP and ILS solutions. For each instance, two scenarios are considered. The first one does not include an uncertainty aspect in the processing time of tasks. In contrast, the second one considers this aspect by modeling the processing time of tasks according to a normal distribution.

Based on the KPIs defined in the performance view, the Table F.28 and the Table F.29 summarize respectively the simulation results of the percentage of workstations in risk with $F_{risk} = 15\%$ and the customer's demand filling rate.

Table F.28 shows that the ILS solution considering ergonomic criterion is more robust than the SALBP solution. Because when considering stochastic task times, this later grows considerably from 35% to 92% of workstations in risk, while the ILS solution reaches only 39%.

The results of Table F.29 highlight the robustness of the ILS solution regarding the SALBP one. For the ILS solution, the customer's demand-filling rate stabilizes around 99% for both deterministic and stochastic task times. Whereas in the case of the SALBP solution, this rate decreases.

Deterministic task times		**Stochastic task times**	
SALBP	ILS	SALBP	ILS
35.71%	7.14%	92.86%	39.86%

Table F.28: % of workstations in risk SALBP and ILS solutions.

Determinist task times		**Stochastic task times**	
SALBP	ILS	SALBP	ILS
98.03%	99.01%	97.63%	99.06%

Table F.29: Customer's demand filling rate for SALBP and ILS solutions.

F.6.5 Maintenance phase

In Figure F.17, we represent the percentage of capacity utilization rate at each workstation, for both SALBP and ILS solutions, in the stochastic scenario. The result does not vary from the working time obtained in Figure F.15. The solution proposed with ILS confirmed its robustness against hazards during the various production cycles.

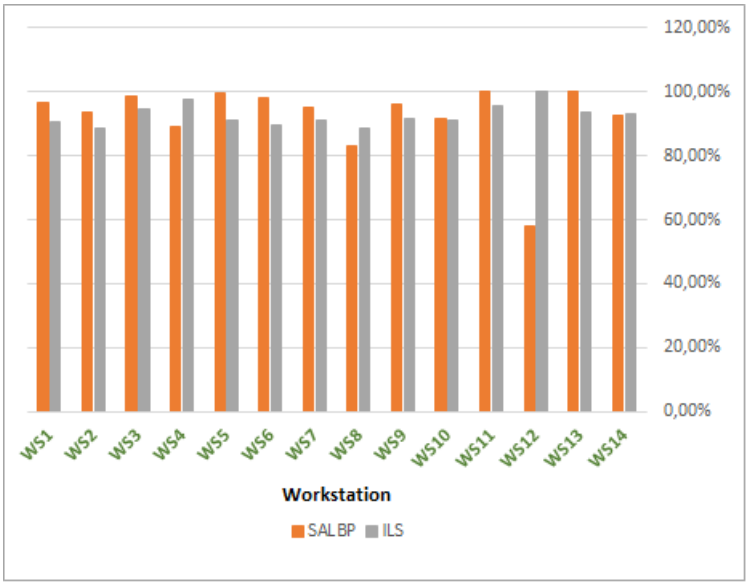

Figure F.17: % of capacity utilization for SALBP and ILS Solutions in the stochastic scenario.

The simulation results have been communicated to the process planning department to discuss their feasibility, and we recommended applying the solution obtained by a mathematical model that uses ergonomic criteria.

Likewise, we update the corresponding documentation and store the new version of the produce module.

F.7 Instantiation of Models

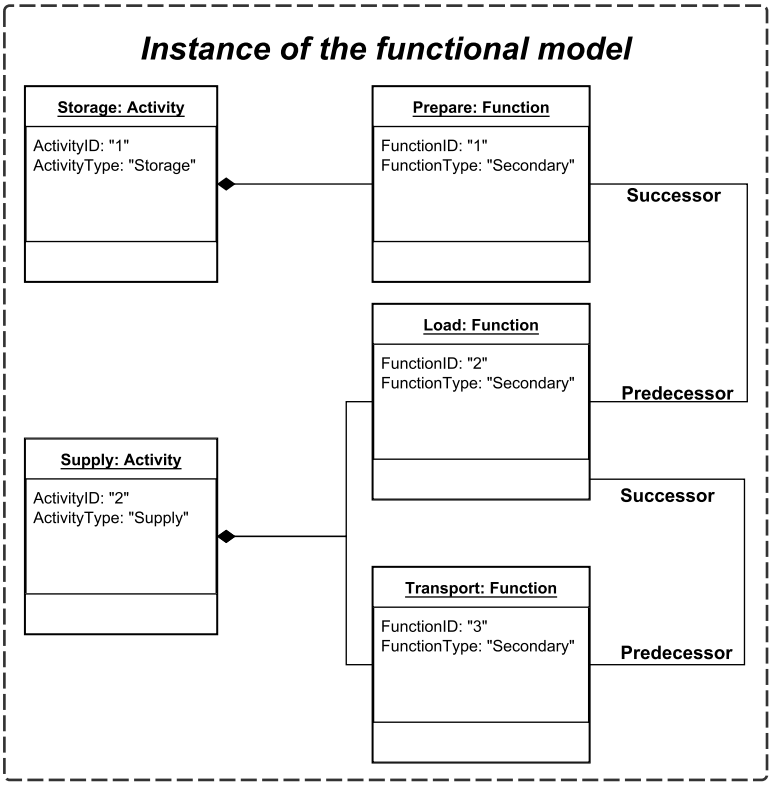

Figure F.18: Instance of the functional model of the use case 1.

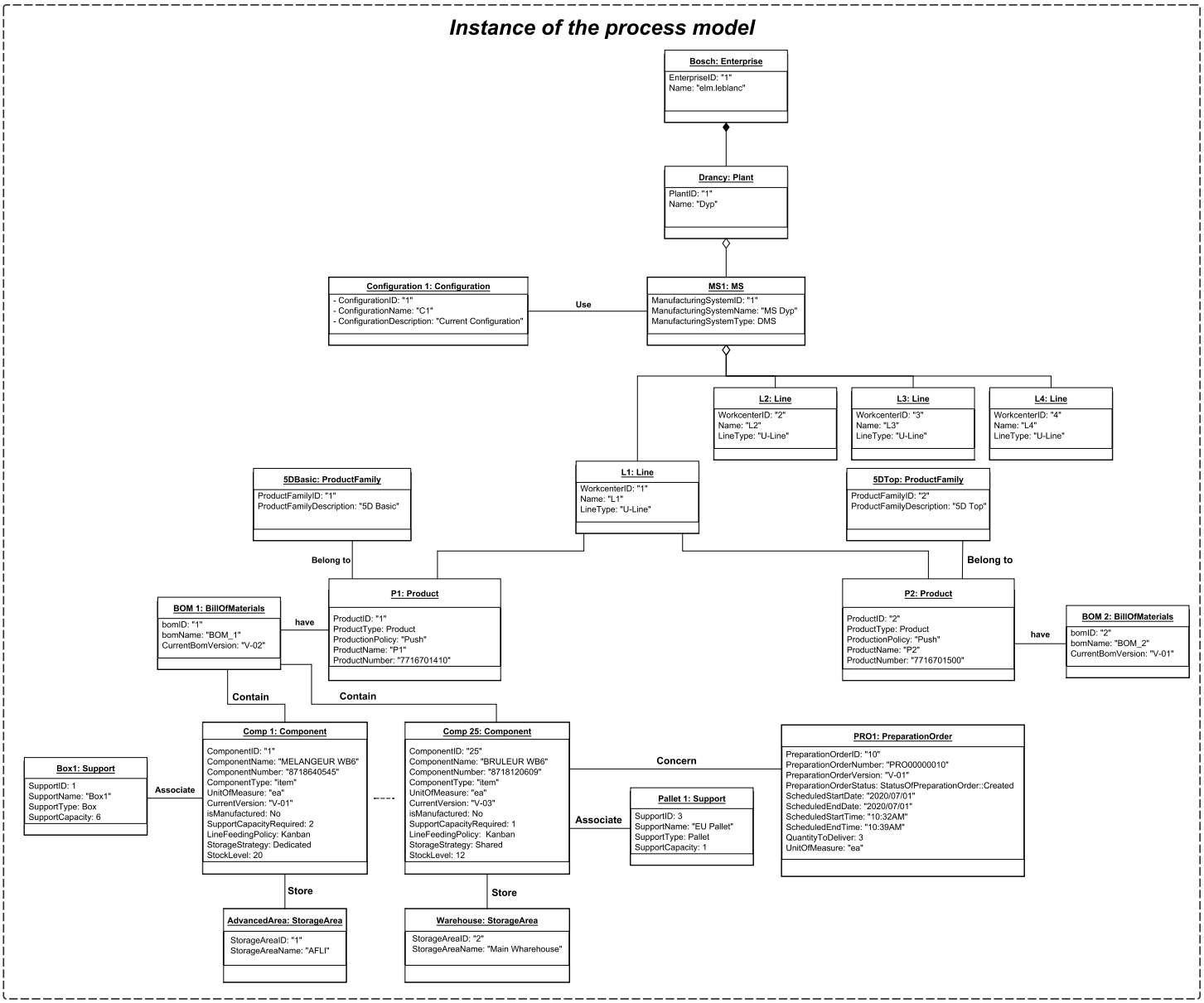

Figure F.19: Instance of the process model of the use case 1.

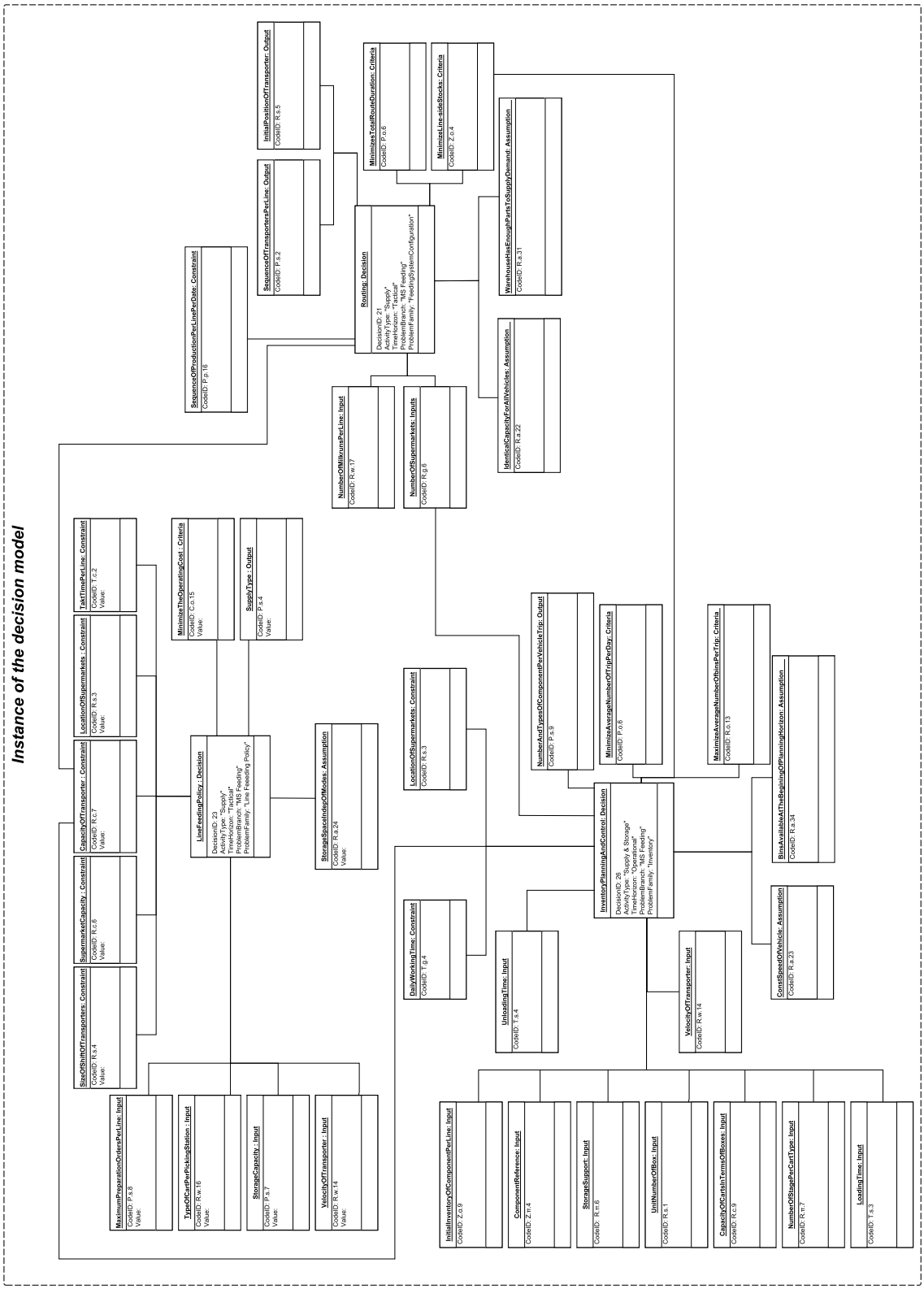

Figure F.20: Instance of the decision model of the use case 1.

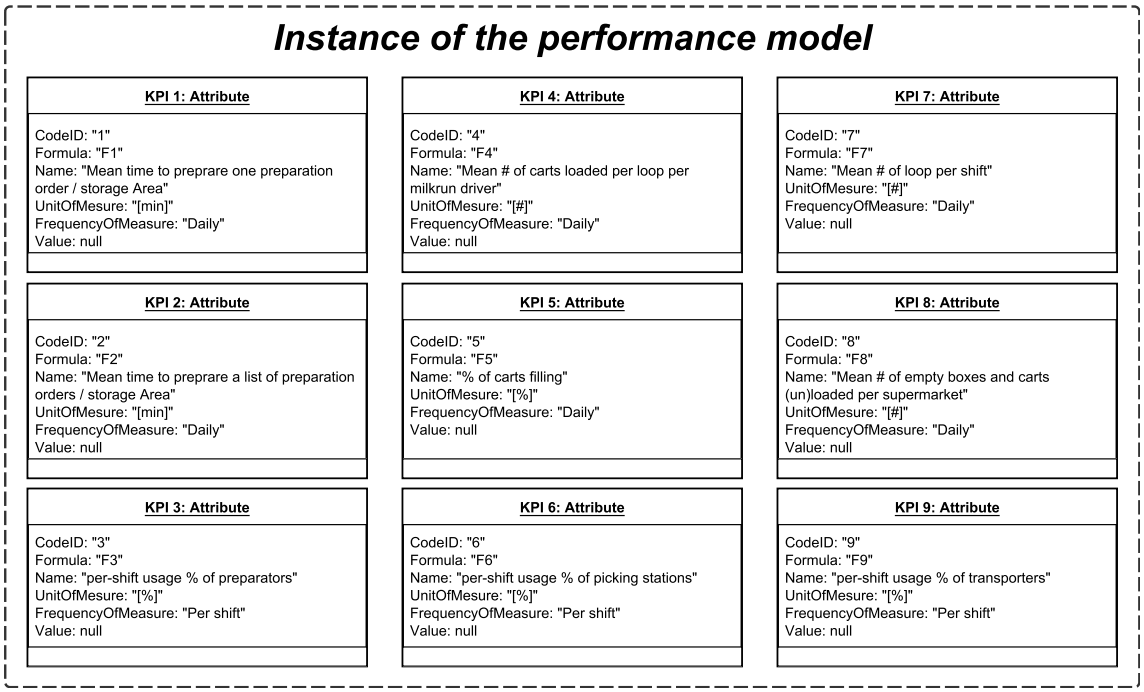

Figure F.21: Instance of the performance model of the use case 1.

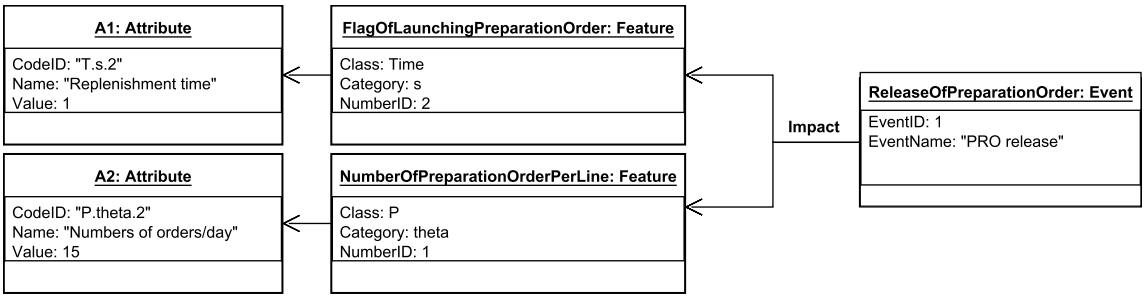

Figure F.22: Instance of the information model of the use case 1.

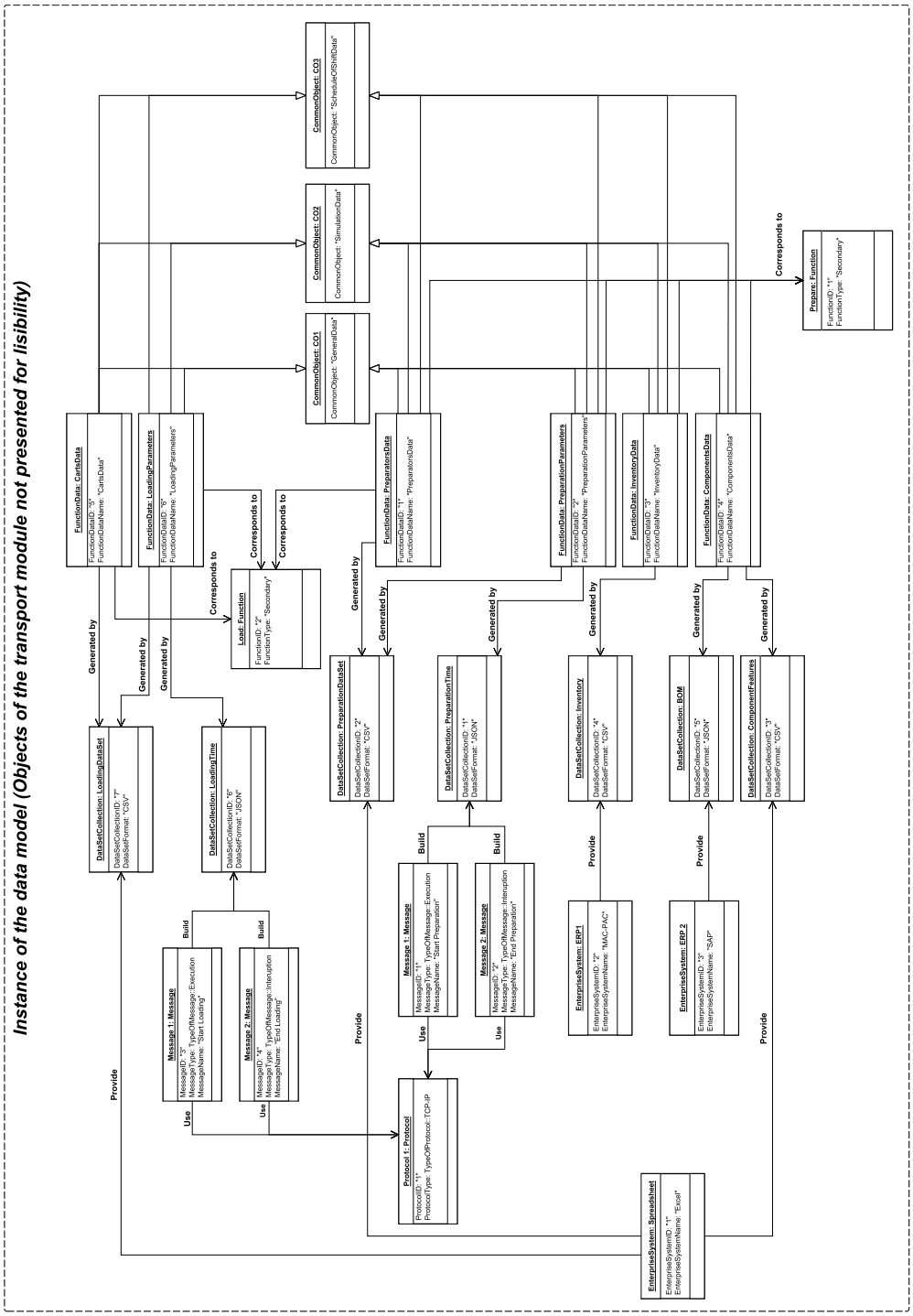

Figure F.23: Instance of the data model of the use case 1.

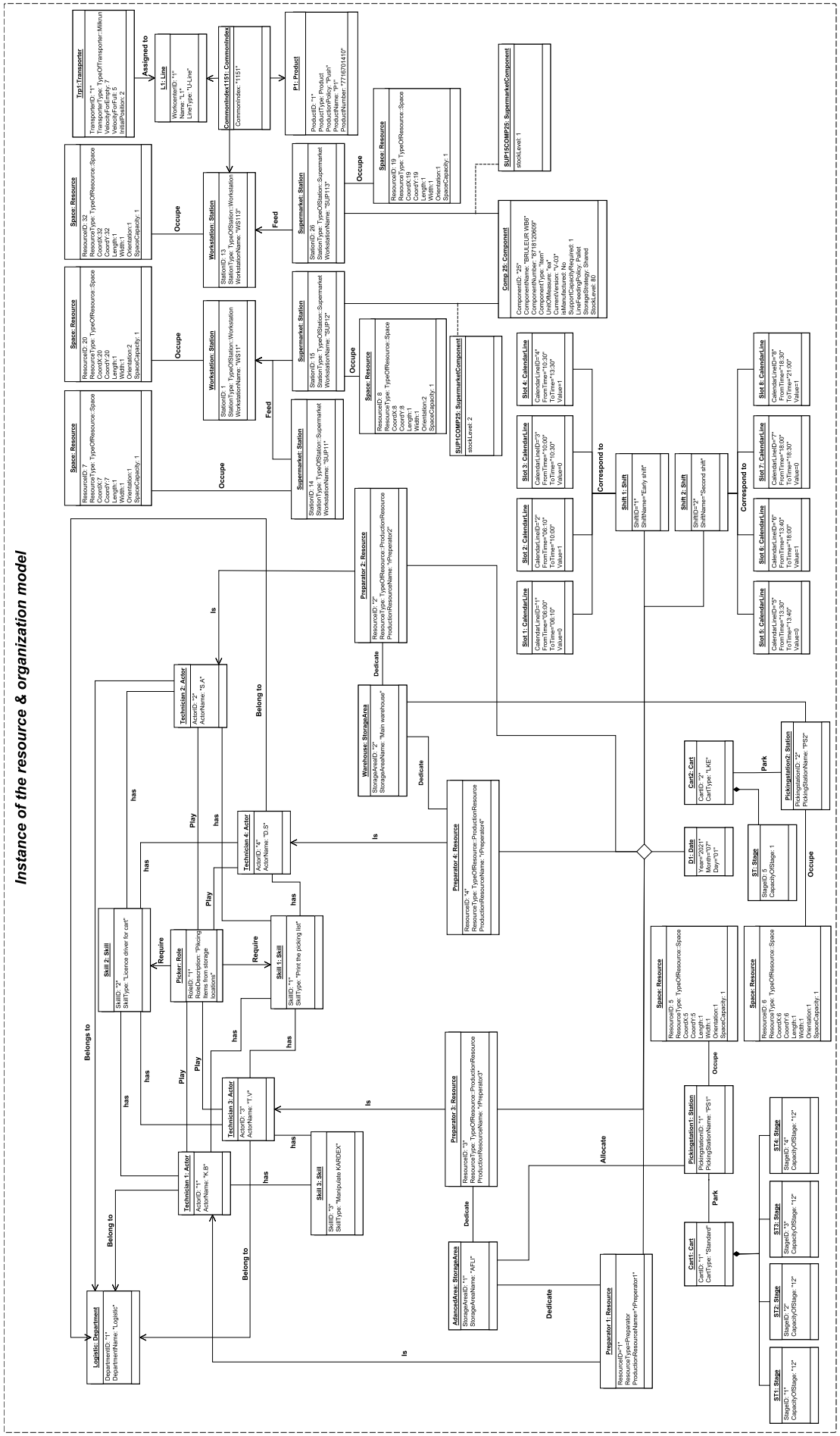

Figure F.24: Instance of the resource & orgranization model of the use case 1.

Appendix G

List of utilities

This appendix defines the classical modeling choices and provide the list of functions useful for the definition of the layers of the development architecture used in the MEMO I4.0 project within elm.leblanc company. It also provides a table of P-values of the Kolmogorov-Smirnov test.

Table G.1: Definition of classical modeling choices.

Modeling choice	**Definition**
Entity	*an object presenting an element of interest* within the modeled system. It represents elements that flow through the simulation model, such as customers, products, or vehicles. Each entity possesses specific attributes and state variables, influencing its behavior and interactions, driving the dynamics of the simulation.
Resource	*an element required to perform specific tasks* within the simulated system. It is crucial in enabling entities to carry out actions and tasks. Resources are associated with attributes such as capacity, availability, and utilization, which determine how and when entities can access and use them. Resources are limited in quantity and can be shared among entities, influencing system performance.
Queue	*a data structure* that temporarily holds entities waiting for service or processing when resources are not immediately available to handle them. Entities enter the queue and wait until a resource is available for processing.
Delay	*an amount of time* an entity spends waiting or idling before continuing its processing or moving to the next stage within the simulation model. It is a concept used to represent various time-related processes, such as service times, processing times, transportation times, and inter-arrival times.
Process	*a specific sequence of actions or tasks* performed by entities within the simulation model. When an event initiates a process, the corresponding actions or tasks are executed, and the state of the involved entities and resources may change accordingly. The interaction of multiple processes within the simulation model defines the overall system behavior and the order in which events occur. Process execution significantly impacts the output and results of the simulation.
Set	*a specific data structure* that organizes and manages entities that share common attributes. It facilitates grouping entities based on shared characteristics, enabling easier access and manipulation during the simulation process.
Expression	*a logical statement* that allows for the computation of values and set conditions within the simulation model. It helps define various simulation aspects, such as entity attributes, resource capacities, process times, and dynamic rules.
Advanced set	*a type of data structure* that offers powerful features and the ability to create and use either sets or expressions, allowing for more sophisticated modeling and simulation of complex systems with custom rules and logic.

```
Public Function getResultOfQueryFromProcedure(ByRef serverName As String, ByRef instanceName As String, ByRef databaseName As String,
ByRef procedureName As String, ByRef parameterName1 As String, ByRef parameterName2 As String, ByVal parameter1 As Long, ByVal parameter2 As Long, _
Optional ByRef temporaryDBName As String, Optional ByRef temporaryTable As String, Optional ByRef nameOfField As String) As Variant

    Dim rst As New ADODB.Recordset
    Dim cnn As New ADODB.Connection
    Dim cnnTemp As New ADODB.Connection
    Dim cnnStr As String
    Dim cnnTempStr As String
    Dim StrSproc As String

    cnnStr = "Provider=MSOLEDBSQL;Server=" & serverName & "\" & instanceName & ";Database=" & databaseName & ";Trusted_Connection=yes;"

    With cnn
        .CommandTimeout = 900
        .ConnectionString = cnnStr
        .Open
    End With

    StrSproc = "EXEC " & procedureName &
            " @" & parameterName1 & "=" & parameter1 & "," & _
            " @" & parameterName2 & "=" & parameter2

    rst.Open StrSproc, cnn

    If temporaryDBName <> "" Then

        'Ask result from
        cnnTempStr = "Provider=MSOLEDBSQL;Server=" & serverName & "\" & instanceName & ";Database=" & temporaryDBName & ";Trusted_Connection=yes;"

        With cnnTemp
            .CommandTimeout = 900
            .ConnectionString = cnnTempStr
            .Open
        End With

        'Store the query in a string
        query = "SELECT " & nameOfField & " FROM [" & temporaryTable & "]"

        ' Run the query and store in a recordset
        rs.Open query, cnnTemp

        ' Write data
        getResultOfQueryFromProcedure = rs.GetRows

    Else

        getResultOfQueryFromProcedure = rst.GetRows

    End If

    'Close the recordsets & Connections
    rs.Close
    cnnTemp.Close
    cnn.Close

    Set rst = Nothing
    Set rs = Nothing
    Set cnn = Nothing
    Set cnnTemp = Nothing

End Function
```

Figure G.1: Function used by DAO to obtain data from the procedures defined in database.

n	α 0.01	α 0.05	α 0.1	α 0.15	α 0.2
1	0.995	0.975	0.950	0.925	0.900
2	0.929	0.842	0.776	0.726	0.684
3	0.828	0.708	0.642	0.597	0.565
4	0.733	0.624	0.564	0.525	0.494
5	0.669	0.565	0.510	0.474	0.446
6	0.618	0.521	0.470	0.436	0.410
7	0.577	0.486	0.438	0.405	0.381
8	0.543	0.457	0.411	0.381	0.358
9	0.514	0.432	0.388	0.360	0.339
10	0.490	0.410	0.368	0.342	0.322
11	0.468	0.391	0.352	0.326	0.307
12	0.450	0.375	0.338	0.313	0.295
13	0.433	0.361	0.325	0.302	0.284
14	0.418	0.349	0.314	0.292	0.274
15	0.404	0.338	0.304	0.283	0.266
16	0.392	0.328	0.295	0.274	0.258
17	0.381	0.318	0.286	0.266	0.250
18	0.371	0.309	0.278	0.259	0.244
19	0.363	0.301	0.272	0.252	0.237
20	0.356	0.294	0.264	0.246	0.231
25	0.320	0.270	0.240	0.220	0.210
30	0.290	0.240	0.220	0.200	0.190
35	0.270	0.230	0.210	0.190	0.180
40	0.250	0.210	0.190	0.180	0.170
45	0.240	0.200	0.180	0.170	0.160
50	0.230	0.190	0.170	0.160	0.150
OVER 50	$\frac{1.63}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.14}{\sqrt{n}}$	$\frac{1.07}{\sqrt{n}}$

Figure G.2: P-values of the Kolmogorov-Smirnov test.

```
Private moduleOfElement As Module
Private nameOfElement As String
Private typeOfElement As String
Private data() As Variant
Private parameters() As Variant

Property Get getModule() As Module
    getModule = moduleOfElement
End Property

Property Let setModule(ByRef elementOfModule As Module)
    moduleOfElement = elementOfModule
End Property

Property Get getNameOfElement() As String
    getNameOfElement = nameOfElement
End Property

Property Let setNameOfElement(ByVal elementName As String)
    nameOfElement = elementName
End Property

Property Get getTypeOfElement() As String
    getTypeOfElement = typeOfElement
End Property

Property Let setTypeOfElement(ByVal elementType As String)
    typeOfElement = elementType
End Property

Property Get getData() As Variant
    getData = data
End Property

Property Let setData(ByRef dataOfElement As Variant)
    data = dataOfElement
End Property

Property Get getParameters() As Variant
    getParameters = parameters
End Property

Property Let setParameters(ByRef parametersOfElement As Variant)
    parameters = parametersOfElement
End Property
```

Figure G.3: Class of Element to create in simulation platform.

```
Public Sub createElement(ByRef listOfMainSet As Variant, ByRef listOfTypeOfMainSet As Variant, ByRef listTypeOfMainSet As Variant, ByRef listTypeOfPanelOfMainSet As Variant, _
ByRef listOfParametersOfMainSet As Variant, ByRef listOfDataOfMainSet As Variant, _
ByRef ListOfElements As Variant, ByRef listTypeOfElements As Variant, ByRef listTypeOfPanelOfElements As Variant, _
ByRef listOfTypeOfElementsType As Variant, ByRef listOfNumberOfElements As Variant, ByRef listOfParametersOfElements As Variant, _
ByRef listOfDataOfElements As Variant, _
ByRef listOfSubElements As Variant, ByRef listTypeOfSubElements As Variant, ByRef listTypeOfPanelOfSubElements As Variant, _
ByRef listOfNumberOfSubElements As Variant, ByRef listOfParametersOfSubElements As Variant, ByRef listOfDataOfSubElements As Variant)

Dim elementObj As New Element
Dim subElementObj As New Element
Dim subSubElementObj As New Element

For i = 1 To UBound(listOfMainSet)

        indexModule = m.Modules.FindByData("Name", listOfMainSet(i))

        If indexModule <> 0 Then

                Set mainModule = m.Modules.Item(indexModule)

        Else
                'Create the corresponding Main Set

                Set mainModule = m.Modules.Create(listTypeOfPanelOfMainSet(i), listTypeOfMainSet(i), 1, 1)
                mainModule.Data("Name") = listOfMainSet(i)
                mainModule.Data("Type") = listOfTypeOfMainSet(i)
                mainModule.UpdateShapes
                indexModule = m.Modules.FindByData("Name", listOfMainSet(i))
                Set mainModule = m.Modules.Item(indexModule)

        End If

        On Error Resume Next

        If ListOfElements(i) <> "" Then

                    ReDim Preserve listNameOfElements(listOfNumberOfElements(i))

                    For j = 1 To listOfNumberOfElements(i)

                        nameOfElement = ListOfElements(i) & j
                        listNameOfElements(j) = nameOfElement

                        'Reading Data And Creating Module

                            'Check if exist SubElement For this Element

                            If listOfSubElements(j) <> "" Then

                                    If j = 3 Then

                                        listOfNumberOfSubElements(j) = nbrLine * UBound(trp.getTransportersData(i, 1), 1)

                                    End If

                                    ReDim Preserve listNameOfSubElements(listOfNumberOfSubElements(j))

                                    'Create the corresponding SubElement

                                        For k = 1 To listOfNumberOfSubElements(j)

                                            indexSubElementModule = m.Modules.FindByData("Name", listOfSubElements(i) & j & k)
```

Figure G.4: Create element in simulation plateform (Part 1).

```
If indexSubElementModule = 0 Then

     'Create the corresponding subElement

    Set subElementModule = m.Modules.Create(listTypeOfPanelOfSubElements(j), listTypeOfSubElements(j), 1, 1)

Else

    'Update the corresponding subElement

     Set subElementModule = m.Modules.Item(indexSubElementModule)

End If

listNameOfSubElements(k) = listOfSubElements(i) & j & k

subSubElementObj.setModule (subElementModule)
subSubElementObj.setNameOfElement (listNameOfSubElements(k))
subSubElementObj.setData (listOfDataOfSubElements(j))

 If subSubElementObj.getTypeOfElement = "Expression" Then

     subSubElementObj.setTypeOfElement ("Native")
     ReDim Preserve subIndexes(1 To 5)

         For l = 1 To 5

             If listOfParametersOfSubElements(k)(l) = "var" Then

                 If l = 1 Or l = 3 Then

                     subIndexes(l) = j

                 ElseIf l = 2 Or l = 4 Then

                     subIndexes(l) = k

                 End If

             Else

                 subIndexes(l) = listOfParametersOfSubElements(k)(l)

             End If


         Next l

     ReDim Preserve subParameters(1 To 5)

    subParameters(1) = UBound(subSubElementObj.GetData(subIndexes(1), subIndexes(2)), 1)
    subParameters(2) = UBound(subSubElementObj.GetData(subIndexes(1), subIndexes(2)), 2)
    subParameters(3) = subIndexes(3)
    subParameters(4) = subIndexes(4)
    subParameters(5) = subIndexes(5)

 Else

     ReDim Preserve Parameters(1 To 1)

     subParameters(1) = listOfParametersOfSubElements(k)(1)
```

Figure G.5: Create element in simulation plateform (Part 2).

```
                                End If

                                subSubElementObj.setParametersOfElements (subParameters)

                                Call assignElementData(subSubElementObj)

                                Erase subParameters
                                Erase subIndexes
                                Set subElementModule = Nothing
                                Set subSubElementObj = Nothing

                            Next k

                            'Create the corresponding Element

                              indexElementModule = m.Modules.FindByData("Name", listNameOfElements(j))

                               If indexElementModule = 0 Then

                                    Set elementModule = m.Modules.Create(listTypeOfPanelOfElements(i), listTypeOfElements(i), 1, 1)

                               Else

                                    'Update the corresponding Element

                                    Set elementModule = m.Modules.Item(indexElementModule)

                               End If

                               subElementObj.setModule (elementModule)
                               subElementObj.setNameOfElement (listNameOfElements(j))
                               subElementObj.setTypeOfElement (listOfTypeOfElementsType(j))
                               subElementObj.setData (listNameOfSubElements)

                                    Call assignElementData(subElementObj)

                                    Err.Clear
                                    Set subElementModule = Nothing
                                    Set elementModule = Nothing
                                    Set subElementObj = Nothing

        Else

                'Create the corresponding Element

                indexElementModule = m.Modules.FindByData("Name", listNameOfElements(j))

                       If indexElementModule = 0 Then

                            Set elementModule = m.Modules.Create(listTypeOfPanelOfElements(i), listTypeOfElements(i), 1, 1)

                       Else

                            'Update the corresponding Element

                            Set elementModule = m.Modules.Item(indexElementModule)

                       End If
```

Figure G.6: Create element in simulation plateform (Part 3).

```
                    subElementObj.setModule (elementModule)
                    subElementObj.setNameOfElement (listNameOfElements(j))
                    subElementObj.setTypeOfElement ("Real")
                    subElementObj.setData (listOfDataOfElements(j))

                    If subElementObj.getTypeOfElement = "Expression" Then

                         subElementObj.setTypeOfElement ("Real")
                         ReDim Preserve indexes(1 To 5)

                             For p = 1 To 5

                                 If listOfParametersOfElements(j)(p) = "var" Then

                                     If p = 1 Or p = 3 Then

                                         indexes(p) = i

                                     ElseIf p = 2 Or p = 4 Then

                                         indexes(p) = j

                                     End If

                                 Else

                                     indexes(p) = listOfParametersOfElements(j)(p)

                                 End If


                             Next p

                         ReDim Preserve Parameters(1 To 5)

                        Parameters(1) = UBound(subElementObj.GetData(indexes(1), indexes(2)), 1)
                        Parameters(2) = UBound(subElementObj.GetData(indexes(1), indexes(2)), 2)
                        Parameters(3) = indexes(3)
                        Parameters(4) = indexes(4)
                        Parameters(5) = indexes(5)

                    Else

                         ReDim Preserve Parameters(1 To 1)

                         Parameters(1) = listOfParametersOfElements(j)(1)

                    End If

                    subElementObj.setParametersOfElements (Parameters)

                    Call assignElementData(subElementObj)

                Err.Clear

                Erase Parameters
                Erase indexes
                Set subElementModule = Nothing
                Set elementModule = Nothing
                Set subElementObj = Nothing

            End If
```

Figure G.7: Create element in simulation plateform (Part 4).

```
                    Next j

                    elementObj.setModule (mainModule)
                    elementObj.setNameOfElement (listOfMainSet(i))
                    elementObj.setTypeOfElement (listOfTypeOfMainSet(i))
                    elementObj.setData (listNameOfElements)

        Else


                    elementObj.setModule (mainModule)
                    elementObj.setNameOfElement (listOfMainSet(i))
                    elementObj.setTypeOfElement (listOfTypeOfMainSet(i))
                    elementObj.setData (listOfDataOfMainSet(j))

                        If elementObj.getTypeOfElement = "Expression" Then

                            elementObj.setTypeOfElement ("Native")

                            ReDim Preserve mainIndexes(1 To 5)

                                For o = 1 To 5

                                    If listOfParametersOfMainSet(i)(o) = "var" Then

                                        mainIndexes(o) = i

                                    Else

                                        mainIndexes(o) = listOfParametersOfMainSet(i)(o)

                                    End If


                                Next o

                            ReDim Preserve mainParameters(1 To 5)

                            mainParameters(1) = UBound(elementObj.GetData(mainIndexes(1), mainIndexes(2)), 1)
                            mainParameters(2) = UBound(subSubElementObj.GetData(mainIndexes(1), mainIndexes(2)), 2)
                            mainParameters(3) = mainIndexes(3)
                            mainParameters(4) = mainIndexes(4)
                            mainParameters(5) = mainIndexes(5)

                        Else

                            ReDim Preserve mainParameters(1 To 1)
                            mainParameters(1) = listOfParametersOfSubElements(i)(1)

                        End If

                        elementObj.setParametersOfElements (mainParameters)
                        Erase mainIndexes
                        Erase mainParameters
        End If


        Call assignElementData(elementObj)
        Set elementObj = Nothing

Next i
End Sub
```

Figure G.8: Create element in simulation plateform (Part 5).

```
Public Sub DeleteElement(ByRef ListOfElements As Variant, ByRef ListOfTypes As Variants, ByRef ListOfSubTypes As Variant, ByRef ListOfLevels As Variant)

Dim m As Model
Dim s As SIMAN

Dim moMainElement As Module
Dim moElement As Module
Dim moSubElement As Module


Dim indexMainElement As Long
Dim indexElement As Long
Dim indexSubElement As Long

Dim nameOfElement As String
Dim nameOfSubElement As String

Dim l As Long
Dim t As Long

Set m = ThisDocument.Model
Set s = m.SIMAN


For i = 1 To UBound(ListOfElements)

    indexMainElement = m.Modules.FindByData("Name", ListOfElements(i))

    If indexMainElement <> 0 Then

        Set moMainElement = m.Modules.Item(indexMainElement)
        l = 1
        indexElement = 1

        If ListOfLevels(i) > 0 Then

                While indexElement <> 0

                    Select Case ListOfTypes(i)

                    Case "Other"

                        nameOfElement = moMainElement.Data("Other(" & l & ")" & "")

                    Case Else

                        nameOfElement = moMainElement.Data(ListOfTypes(i) & " Name(" & l & ")" & "")

                    End Select

                    indexElement = m.Modules.FindByData("Name", nameOfElement)
                    Set moElement = m.Modules.Item(indexElement)

                    t = 1
                    indexSubElement = 1

                    If ListOfLevels(i) > 1 Then

                        While indexSubElement <> 0

                            Select Case ListOfSubTypes(i)
```

Figure G.9: Delete element in simulation plateform (Part 1).

```
                    If ListOfLevels(i) > 1 Then

                        While indexSubElement <> 0

                            Select Case ListOfSubTypes(i)

                                Case "Other"

                                    nameOfSubElement = moElement.Data("Other(" & t & ")" & "")

                                Case Else

                                    nameOfSubElement = moElement.Data(ListOfSubTypes(i) & " Name(" & t & ")" & "")

                            End Select

                            indexSubElement = m.Modules.FindByData("Name", nameOfSubElement)
                            Set moSubElement = m.Modules.Item(indexSubElement)
                            moSubElement.Shape.Selected = True
                            m.ActiveView.Selection.Delete

                            t = t + 1

                        Wend

                    End If

                    Err.Clear
                    moElement.Shape.Selected = True
                    m.ActiveView.Selection.Delete
                    l = l + 1

                Wend
            End If

            Err.Clear
            moMainElement.Shape.Selected = True
            m.ActiveView.Selection.Delete

            Set moMainElement = Nothing
            Set moElement = Nothing
            Set moSubElement = Nothing

        End If

    Next i

    End Sub
```

Figure G.10: Delete element in simulation plateform (Part 2).

```
Public Sub assignElementData(ByRef elementObj As Element)
    Select Case elementObj.getTypeOfElement
    Case "Advanced Set"
        Call assignElementsOfAdvancedSet(elementObj.getModule, elementObj.getTypeOfElement, elementObj.getData)
    Case "Expression"
        Call assignElementsOfExpression(elementObj.getModule, elementObj.getNameOfElement, elementObj.getTypeOfElement, elementObj.getData, elementObj.getParameters)
    Case "Queue"
        Call assignElementOfQueue(elementObj.getModule, elementObj.getNameOfElement, elementObj.getTypeOfElement, elementObj.getParameters)
    Case "Set"
        Call assignElementsOfSet(elementObj.getModule, elementObj.getNameOfElement, elementObj.getTypeOfElement, elementObj.getData, elementObj.getParameters)
    Case "Sequence"
        Call assignElementsOfSequence(elementObj.getModule, elementObj.getData)
    Case Else
        MsgBox "Aucun Type correspondant"
    End Select
End Sub
```

Figure G.11: Assignement of data according to element type.

```
Public Sub assignElementsOfAdvancedSet(ByRef asMod As Module, ByRef typeOfAS As String, ByRef elementListOfAS As Variant)
On Error Resume Next
    If typeOfAS = "Other" Then
        For i = 1 To UBound(elementListOfAS)
            asMod.data("Other(" & i & ")" & "") = elementListOfAS(i)
        Next i
    ElseIf typeOfAS = "Queue" Then
        For i = 1 To UBound(elementListOfAS)
            asMod.data("Queue Name(" & i & ")" & "") = elementListOfAS(i)
        Next i
    End If
    asMod.UpdateShapes
End Sub
```

Figure G.12: Assignement of data to advanced set element.

```
Public Sub assignElementsOfExpression(ByRef exMod As Module, ByRef nameOfEX As String, ByRef typeOfData As String, ByRef elementListOfEX As Variant, ByRef parametersOfExpressions As Variant)

    exMod.data("Name") = nameOfEX
    exMod.data("Dim1") = parametersOfExpressions(1)
    exMod.data("Data Type") = typeOfData

    If parametersOfExpressions(2) <> 0 Then

        exMod.data("Dim2") = parametersOfExpressions(2)

                For p = 1 To parametersOfExpressions(2)

                    For r = 1 To parametersOfExpressions(1)

                        If parametersOfExpressions(3) <> 0 Then

                            If parametersOfExpressions(4) <> 0 Then

                                exMod.data("Value(" & parametersOfExpressions(1) * (p - 1) + r & ")") = elementListOfEX(parametersOfExpressions(3), parametersOfExpressions(4))(r, p)

                            Else

                                exMod.data("Value(" & parametersOfExpressions(1) * (p - 1) + r & ")") = elementListOfEX(parametersOfExpressions(3), parametersOfExpressions(5))(r, p)

                            End If

                        Else

                                If parametersOfExpressions(5) <> 0 Then

                                    exMod.data("Value(" & r & ")") = elementListOfEX(r, parametersOfExpressions(5))

                                Else

                                    exMod.data("Value(" & parametersOfExpressions(1) * (p - 1) + r & ")") = elementListOfEX(r, p)

                                End If

                        End If

                    Next r

                Next p

    Else

                 For r = 1 To parametersOfExpressions(1)

                        If parametersOfExpressions(5) <> 0 Then

                            exMod.data("Value(" & r & ")") = elementListOfEX(r, parametersOfExpressions(5))

                        End If

                  Next r


    End If
    exMod.UpdateShapes
End Sub
```

Figure G.13: Assignement of data to expression element.

```
Public Sub assignElementOfQueue(ByRef qMod As Module, ByRef nameOfQueue As String, ByRef typeOfQueue As String, ByRef parametersOfQueue As Variant)

On Error Resume Next

    qMod.data("Name") = nameOfQueue
    qMod.data("Type") = typeOfQueue

    If typeOfQueue = "Lowest Attribute Value" Or typeOfQueue = "Highest Attribute Value" Then

            qMod.data("Attribute") = parametersOfQueue(1)
    End If

    qMod.UpdateShapes

End Sub
```

Figure G.14: Assignement of data to queue element.

```
Public Sub assignElementsOfSequence(ByRef seqMod As Module, ByRef elementListOfSEQ As Variant)

On Error Resume Next

        For i = 1 To UBound(elementListOfSEQ)

            seqMod.data("Station(" & i & ")" & "") = elementListOfSEQ(i)

        Next i

   seqMod.UpdateShapes

End Sub
```

Figure G.15: Assignement of data to sequence element.

```
Public Sub assignElementsOfSet(ByRef sMod As Module, ByRef nameOfSET As String, ByRef typeOfSET As String, ByRef elementListOfSET As Variant, ByRef parametersOfSet As Variant)

On Error Resume Next

    sMod.data("Name") = nameOfSET
    sMod.data("Type") = typeOfSET
    sMod.data("MemDef") = parametersOfSet(1)

    If typeOfSET = "Resource" Then

        For i = 1 To UBound(elementListOfSET)

            sMod.data("MemSet(" & i & ")" & "") = "Single Element"
            sMod.data("Resource Name(" & i & ")" & "") = elementListOfSET(i)

        Next i

    ElseIf typeOfSET = "Entity Picture" Then

        For i = 1 To UBound(elementListOfSET)

            sMod.data("Picture Name(" & i & ")" & "") = elementListOfSET(i)

        Next i

    ElseIf typeOfSET = "Tally" Then

        For i = 1 To UBound(elementListOfSET)

            sMod.data("Tally Name(" & i & ")" & "") = elementListOfSET(i)

        Next i

    ElseIf typeOfSET = "Counter" Then

        For i = 1 To UBound(elementListOfSET)

            sMod.data("Counter Name(" & i & ")" & "") = elementListOfSET(i)

        Next i

    End If

    sMod.UpdateShapes

End Sub
```

Figure G.16: Assignement of data to set element.

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : 2023EMSEM032

Mahdi EL ALAOUI EL ABDELLAOUI

MÉTHODOLOGIE DE MODÉLISATION D'ENTREPRISE ORIENTÉE 4.0: APPLICATION AUX SYSTÈMES D'AIDE À LA DÉCISION POUR LES SYSTÈMES PRODUCTION

Spécialité : Génie Industriel

Industrie 4.0, Usine Intelligente, Architecture d'enteprise, Modélisation d'entrepirse, Systèmes de production, Jumeau numérique, Simulation, Optimisation, Systèmes d'aide à la décision, Nouveaux indicateurs clés de performance

Résumé :

La nouvelle révolution industrielle «Industrie 4.0 »complexifie la prise de décision dans les systèmes de production en raison de nombreux défis, tels que le cycle de vie court du produit, l'intégration des données et la considération de la durabilité. Les entreprises ont besoin d'aide pour relever ces défis et déployer une usine intelligente.

Cette thèse s'inscrit dans ce cadre, et propose une nouvelle méthodologie d'implémentation des systèmes d'aide à la décision orientée Industrie 4.0. Tout d'abord, nous identifions et classons les problèmes de décision d'un système de production, et examinons leurs interactions. Ensuite, nous étudions leurs principales évolutions en Industrie 4.0, et proposons une cartographie permettant de les représenter.

Sur cette base, en ayant recours à la modélisation d'entreprise, nous cherchons à développer des systèmes d'aide à la décision agiles et rapidement reconfigurables adaptés à l'industrie 4.0. Par ailleurs, nous identifions des lacunes des cadres de modélisation d'entreprise existantes, et proposons une nouvelle méthodologie de modélisation d'entreprise orientée Industrie 4.0 «MEMO I4.0».

MEMO I4.0 propose une approche structurée de deux étapes. L'étape stratégique/tactique pour créer les modules de base et enrichir la bibliothèque des modules. L'étape opérationnelle pour construire le modèle intégré et évaluer les performances. Enfin, l'application de MEMO I4.0 sur des études de cas industriel de l'entreprise «elm.leblanc» afin de dériver des systèmes d'aide à la décision, a montré sa capacité à réduire considérablement le temps de développement des projets de modélisation, ainsi sa compatibilité avec l'Industrie 4.0.

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : 2023EMSEM032

Mahdi EL ALAOUI EL ABDELLAOUI

METHODOLOGY OF ENTERPRISE MODELING ORIENTED INDUSTRY 4.0: APPLICATION FOR DECISION SUPPORT SYSTEMS IN MANUFACTURING SYSTEMS

Specialty : Industrial Engineering

Keywords : Industry 4.0, Smart manufacturing, Enterprise architecture, Enterprise modeling, Manufacturing systems, Digital twin, Simulation, Optimization, Decision support systems, New key performance indicators

Abstract :

The new industrial revolution, "Industry 4.0" complicates the decision making in manufacturing systems due to many challenges, such as short product life cycle, data integration, and consideration of sustainability. Companies need help to overcome these challenges and deploy a smart factory.

This Ph.D. falls within this context and proposes a new methodology for implementing Industry 4.0-oriented decision support systems. First, we identify and classify the decision problems of a manufacturing system and examine their interactions. Then, we study their main evolutions in Industry 4.0 and propose cartography to represent them.

On this basis, using enterprise modeling, we aim to develop agile and rapidly reconfigurable decision support systems adapted to Industry 4.0. Furthermore, we identify gaps in existing enterprise modeling frameworks and propose a new Industry 4.0-oriented enterprise modeling methodology called «MEMO I4.0».

MEMO I4.0 offers a structured two-step approach. The strategic/tactical step is to create the basic modules and enrich the library of modules. The operational step is to build the integrated model and evaluate performance. Finally, the application of MEMO I4.0 on industrial case studies of the company «elm.leblanc» to derive decision support systems has shown its ability to considerably reduce the development time of modeling projects and its compatibility with Industry 4.0.