Université de Valenciennes et du Hainaut-Cambrésis

**Thèse de doctorat**

**Pour obtenir le grade de Docteur de l'Université de**

**VALENCIENNES ET DU HAINAUT-CAMBRESIS**

Sciences et Technologie, Mention : Informatique :

# Définition et test d'un outil d'aide à l'évaluation de systèmes à architecture à base d'agent, par approche à base de connaissances

**Présentée et soutenue par Selem CHARFI**

**Le 15/02/2013, à Valenciennes**

**Ecole doctorale :**

Sciences Pour l'Ingénieur (SPI)

**Equipe de recherche, Laboratoire :**

Décision, Interaction et Mobilité (DIM)

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH UMR CNRS 8201)

## JURY

**Président du jury**

– BASTIEN Christian. Professeur des Universités. Université de Lorraine

**Rapporteurs**

– GIRARD Patrick. Professeur des Universités. Université de Poitiers
– POIRIER Franck. Professeur des Universités. Université de Bretagne-Sud

**Examinateurs**

– KOLSKI Christophe. Professeur des Universités. Université de Valenciennes et du Hainaut-Cambrésis.

**Directeur de thèse**

– EZZEDINE Houcine. Professeur des Universités. Université de Valenciennes et du Hainaut-Cambrésis

**Co-directeur de thèse**

– MOUSSA Faouzi. Maître de Conférences (HDR). Université de Tunis, Tunisie

*« Une fois résolu,*
*un problème est d'une simplicité atterrante »*

Paolo Cohelo

**Résumé**

Ce mémoire contribue à la thématique de l'évaluation des interfaces utilisateur. Il se base sur deux contributions pour l'évaluation de ces interfaces. La première consiste en un Framework se basant sur trois différentes techniques d'évaluation (le mouchard électronique, le questionnaire et l'inspection des règles ergonomiques). La seconde consiste en deux ensembles de composants graphiques pour la composition d'interfaces graphiques avec les environnements de développement intégré du type WYSIWYG. Le premier ensemble vise à inspecter la présentation statique de l'interface lors de la conception de cette dernière. Cette inspection s'effectue localement par le composant graphique lui-même par rapport à un ensemble de règles ergonomiques. Le second ensemble consiste à l'évaluation de l'interface utilisateur en analysant l'interaction entre l'utilisateur et l'interface. Cet ensemble de composants opère avec un mouchard dédié à cet usage. Une évaluation de ces deux contributions ainsi que des perspectives de recherche concluront ce présent mémoire.

**Mots-Clés.** Interfaces Homme-Machine, Interface Utilisateur, Interaction Homme-Machine, évaluation, techniques d'évaluation, mouchard électronique, règle ergonomique.

---

**Abstract**

This thesis report is a contribution to the user interface evaluation theme. Our thesis is based mainly on two contributions for user interface evaluation. The first one consists of a Framework dedicated for user interface evaluation. It is based on three evaluation techniques (the electronic informer, the questionnaire and the ergonomic inspection). The second consists of two sets of graphical controls for composing the user interface using WYSIWYG integrated development environment. The first set is intended to inspect the user interface static display during the design phase. This inspection is elaborated by the control itself according to ergonomic guidelines. The second set consists in evaluating user interfaces by analyzing the interaction between the user and the interface. This set of controls operates with an electronic informer dedicated to user interface evaluation. An evaluation of these two contributions as well as research perspectives concludes this present report.

**Keywords.** Human-Computer Interaction, User interface, evaluation, evaluation techniques, electronic informer, ergonomic guidelines.

# *Remerciments*

*Enfin rédigée ! ! !*

La rédaction de cette partie est synonyme de « *terminus* » lors du parcours de la thèse. Ainsi, avec une grande joie et beaucoup de soulagements que je m'attaque à cette section. Il s'agit d'un moment attendu depuis fort bien longtemps. Néanmoins, je m'aperçois que cette partie est probablement la plus ardue à rédiger dans ce présent manuscrit. D'une part les lignes semblent de loin assez exigües pour bien remercier tous ceux qui m'ont aidé à achever cette présente thèse. D'autre part, je me trouve face à l'inquiétude d'oublier quiconque des personnes qui, de loin ou de près, m'ont aidé pour achever cette thèse et m'ont supporté pendant ces trois ans (et 3 mois).

J'aimerai adresser mes remercîments notamment à :

- M. **Franck Poirier** (Professeur des Universités à l'*Université de Bretagne-Sud*) et M. **Patrick Girard** (Professeur des Universités à l'*Université de Poitiers*) pour avoir accepté d'être les rapporteurs de cette présente thèse.
- M. **Christian Bastian** (Professeur des Universités à l'*Université de Lorraine*) pour avoir accepté d'être examinateur de cette présente thèse et pour les nombreuses remarques qu'il a faites lors de la soutenance de thèse
- M. **Houcine Ezzedine** (Professeur des Universités à l'*Université de Valenciennes et du Hainaut-Cambrésis*) pour avoir accepté d'assurer le suivi de cette thèse. Toute ma reconnaissance pour la qualité de son encadrement, sa disponibilité, son aide précieuse et ses conseils judicieux. Autre que les compétences de recherche et de rédaction, il m'a cédé un savoir précieux relatif au professionnalisme, à la rigueur et au sens de l'organisation.
- M. **Faouzi Moussa** (Maître de conférences à l'*Université de Tunis*), pour avoir accepté de co-encadrer cette thèse. Je le remercie aussi pour sa disponibilité, ses remarques, ses conseils tout au long de la thèse ainsi que les discussions variées et intéressantes que nous avons eues.
- M. **Christophe Kolski** (Professeur des Universités à l'*Université de Valenciennes et du Hainaut-Cambrésis*), directeur du groupe de recherche *RAIHM*, qui m'a accueilli dans son équipe. Je tiens à lui témoigner ma gratitude pour son aide, ses conseils et surtout ses remarques. Je tiens fortement à le remercier pour ses commentaires toujours aussi précis que pertinents ainsi que pour la lucidité des jugements qu'il a portés à de nombreuses reprises sur mes travaux.

- A ma famille : un grand Merci pour ma famille, mon père **Kamel**, ma mère **Hédia** qui m'ont tant soutenu indéfectiblement, encouragé, aidé, qui m'ont fourni tout ce que je nécessitais et même plus et qui m'ont donné la possibilité de poursuivre mes études, mes deux soeurs **Rawia** et **Sana**, mon beau frère/frère **Hassene** et mon oncle **Anouar**. Je n'arrive pas à trouver les mots convenants pour leur témoigner ma gratitude et mes remercîments. Ils n'ont jamais cessé de me soutenir. Mais je vais me contenter d'un merci (sous prétexte faute de temps et d'espace).
- M. **Abdelwaheb Trabelsi** (Maître Assistant à l'*Université de Sfax*) pour ses conseils très précieux, son aide et son soutien que ça soit lors du parcours de la thèse, du master ou mes études à la faculté de Sciences de Sfax.
- Mes collègues à l'*Université de Valenciennes et du Hainaut-Cambrésis* et notamment : **Abir**, **Amir**, **Anis**, **Clémentine**, **Emine**, **Gael** et **Yoan**. Merci pour votre soutien, bonne humeur, vos discussions enrichissantes et surtout surtout pour les nombreux cafés que vous m'avez payés.
- Un grand merci particulier à mes chers collègues, compagnons de parcours et fidèles amis : **Firas** et **Utku**.
- Mes amis du « *ch'Nord* » avec qui j'ai découvert la région du *Nord Pas de Calais* et partagé des moments inoubliables. Je cite notamment : **Amélie**, **Céline**, **Christelle**, **Gautier**, **Giuliana**, **Jean Michel**, **Magalie**, **Marilyne**, **Natalie** et **Pierre**.
- Mes amis les tunisiens les « expatriés » : **Ahmed**, **Aimen**, **Akram**, **Anis**, **Hammadi**, **Imen**, **Soufiene** et **Wadii**.
- Mes amis en Tunisie : **Anis**, **Fatma**, **Hatem**, **Houcem**, **Omar**, **Mohamed**, **Sabri** et **Walid**.
- Aux enseignants et non enseignants de l'*ISTV de Valenciennes* et *l'IUT, Département Informatique de Maubeuge* où j'ai effectué mes deux contrats d'ATER. Pendant les années de la thèse j'ai intervenu dans certaines matières à l'université. Ceci m'a permis d'acquérir une certaine expérience et d'avoir du recul par rapport à ma thèse.
- La famille **Sellami** pour leur soutien ainsi que pour son aide lors de mon arrivée en France.
- Aux personnes que j'ai oubliées de citer dans cette section et qui auront la bien vaillance de ne pas se fâcher contre moi suite à mon oubli.

PS : Chaque fois ou je cite une liste de personnes, je les cite suivant l'ordre alphabétique pour éviter d'évoquer un ordre de « préférence » ....

# Table des matières

**Abstract** **2**

**Remerciments** **4**

**Liste des Figures** **12**

**Liste des Tableaux** **15**

**Abréviations** **16**

**Introduction générale** **1**

**I L'existant** **3**

**1 L'évaluation des systèmes interactifs vue sous l'angle de l'IHM** **4**
1.1 Concept de l'évaluation des *IHM* . . . . . 5
1.1.1 Objectifs de l'évaluation . . . . . 6
1.1.2 Processus de l'évaluation . . . . . 7
1.2 Techniques d'évaluation . . . . . 9
1.2.1 Classifications existantes . . . . . 12
1.2.2 Proposition de classification des techniques d'évaluation . . . . . 15
1.3 Les techniques d'évaluation . . . . . 16
1.3.1 Classification adoptée . . . . . 16
1.3.2 Format . . . . . 17
1.3.3 Panorama des techniques d'évaluation . . . . . 17
1.3.3.1 Techniques relevant de la classe d'enquête . . . . . 17
1.3.3.2 Techniques relevant de la classe d'inspection de l'utilisabilité . . . . . 20
1.3.3.3 Techniques de test des systèmes interactifs . . . . . 23
1.3.3.4 Techniques de prototypage . . . . . 25
1.3.4 Discussion . . . . . 26
1.4 Difficultés de l'évaluation de l'*IHM* . . . . . 27

1.5 Synthèse et conclusion . . . . . 28

**2 Etat de l'art des outils d'évaluation des systèmes interactifs exploitant les connaissances ergonomiques 30**

2.1 Des notions sur l'ergonomie . . . . . 31
2.1.1 L'ergonomie . . . . . 31
2.1.2 Règle ergonomique . . . . . 32
2.1.3 Critère ergonomique . . . . . 32
2.1.4 Evaluation de la qualité ergonomique des systèmes interactifs . . . 33
2.2 Classification des outils d'évaluation des systèmes interactifs exploitant les connaissances ergonomiques . . . . . 36
2.2.1 Les outils dans lesquels les connaissances ergonomiques sont codées en dur . . . . . 38
2.2.2 Outils dans lesquels les connaissances ergonomiques ne sont pas codées en dur . . . . . 38
2.3 Etat de l'art des outils d'évaluation des *IHM* exploitant les connaissances ergonomiques . . . . . 39
2.3.1 Les outils dont la connaissance ergonomique n'est pas codée en dur 41
2.3.1.1 SYNOP (Kolski et Millot, 1991) . . . . . 41
2.3.1.2 Ocawa : Outil de Contrôle et d'Analyse pour le Web Accessible (Ocawa, 2011) . . . . . 41
2.3.1.3 TAW : Test de Accesibilidad Web (TAW, 2011) . . . . . 42
2.3.1.4 Sherlock (Grammenos *et al.*, 2000) . . . . . 43
2.3.1.5 ERGOVAL (Farenc *et al.*, 2001) . . . . . 44
2.3.1.6 Destine : Design and Evaluation STudio for INtent based Ergonomic web sites (Beirekdar *et al.*, 2002; Beirekdar, 2004) . . . . . 45
2.3.1.7 MAGENTA : Multi-Analysis of Guidelines by an Enhanced Tool for Accessibility (Leporini et Paterno, 2004; Leporini *et al.*, 2006) . . . . . 46
2.3.1.8 L'outil de Takata (Takata *et al.*, 2004) . . . . . 47
2.3.1.9 EvalAccess (Abascal *et al.*, 2004) . . . . . 47
2.3.1.10 WAEX : Web Accessibility Evaluator in a single Xslt file (Centeno *et al.*, 2006, 2007; Freire et de Mattos Fortes, 2005) . . . . . 48
2.3.2 Les outils codant en dur les connaissances ergonomiques . . . . . 48
2.3.2.1 CHIMES : Computer Human Interaction ModElS (Jiang *et al.*, 1992) . . . . . 48
2.3.2.2 KRI/AG : Knowledge-based Review of user Interfaces (Lowgren et Nordqvist, 1992) . . . . . 49
2.3.2.3 Outil d'évaluation des menus de GIBBONS (Gibbons, 1992) . . . . . 49
2.3.2.4 Weblint (Bowers, 1996) . . . . . 50
2.3.2.5 SHERLOCK (Mahajan et Shneiderman, 1997) . . . . . 50
2.3.2.6 HyperAT : Hyperpertext Authoring Tool (Theng et Marsden, 1998) . . . . . 51
2.3.2.7 Bobby (Cooper, 1999) . . . . . 51
2.3.2.8 Dr Watson (Watson, 2011) . . . . . 52

2.3.2.9 Web Tango : Tool for Assessing NaviGation and Organization (Ivory et Hearst, 2001) . . . 52
2.3.2.10 ReWeb et TestWeb (Ricca et Tonella, 2000) . . . 52
2.3.2.11 A-Prompt (A-Prompt, 2011; ATRC, 2002) . . . 53
2.3.2.12 AccessEnable (Brinck et Hofer, 2002) . . . 53
2.3.2.13 WebXACT (Watchfire, 2010) . . . 54
2.3.2.14 HTML Toolbox (NetMechanic, 2011) . . . 54
2.3.2.15 LIFT Machine (Group, 2011) . . . 54
2.3.2.16 WebSAT : Web StatiqueAnalyzer Tool (NIST, 2011) . . . 55
2.3.2.17 Google Analytics (Google, 2011) . . . 55
2.3.2.18 AWebHUT : Automated Web Home-Page Usability Tester (Rukshan et Baravalle, 2011) . . . 56
2.3.2.19 ErgoCoIn (Morandini *et al.*, 2011) . . . 56
2.4 Synthèse et discussion . . . 57

**3 Etat de l'art des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'interface et l'utilisateur 62**
3.1 Aspects considérés pour la présentation des outils d'évaluation . . . 63
3.2 Les outils d'évaluation . . . 65
3.2.1 EMA (Balbo, 1996) . . . 65
3.2.2 USINE (Lecerof et Paterno, 1998) . . . 66
3.2.3 RemUsine (Paterno et Ballardin, 2000) . . . 67
3.2.4 Wet (Etgen et Cantor, 1999) . . . 68
3.2.5 iBot (Zettlemoyer *et al.*, 1999) . . . 69
3.2.6 Web Quilt (Hong *et al.*, 2001; Waterson, May 2002) . . . 70
3.2.7 SMT (Speedzinemedia, 2012) . . . 70
3.2.8 WebRemUsine (Paganelli et Paterno, 2002, 2003) . . . 71
3.2.9 MESIA (Trabelsi, 2006; Trabelsi et Ezzedine, 2012) . . . 72
3.2.10 MM WebRemUsine (Paterno *et al.*, 2007a) . . . 73
3.2.11 Click Tale (C.E.Analytics, 2012) . . . 73
3.2.12 Multidevice WebRemUSINE (Paterno *et al.*, 2007b) . . . 74
3.2.13 Gui Capture and Replay Tool (Jovic *et al.*, 2010; Jovic et Hauswirth, 2010) . . . 75
3.2.14 Click Heat (Labsmedia, 2012) . . . 75
3.2.15 Morae (TechSmith, 2012) . . . 76
3.2.16 EISEval (Tran *et al.*, 2008) . . . 77
3.2.17 Feng-Gui (Shui, 2012) . . . 79
3.2.18 Crazy Egg (CrazyEgg, 2012) . . . 80
3.2.19 Clixpy (Clixpy, 2007) . . . 81
3.2.20 Open Hallway (Hallway, 2012) . . . 82
3.2.21 Usabilla (Usabilla, 2012) . . . 82
3.2.22 Userfly (Userfly, 2009) . . . 83
3.2.23 EVALYZER (Insydelabs, 2013) . . . 84
3.2.24 Loop11 (Loop11, 2012) . . . 84
3.2.25 Silverback (Ltd, 2012) . . . 85
3.3 Synthèse et discussion . . . 86
3.4 Conclusion . . . 89

**II Contribution 90**

**4 Contribution 1 : Le Framework RITA pour l'évaluation des interfaces utilisateurs 91**
4.1 Présentation générale du framework . . . 92
4.1.1 Motivation . . . 92
4.1.2 Concept . . . 92
4.2 Architecture fonctionnelle du framework . . . 93
4.2.1 Module 1 : Gestionnaire de règles ergonomiques . . . 93
4.2.2 Module 2 : le module de capture . . . 96
4.2.2.1 Outil de génération de questionnaire . . . 96
4.2.2.2 Un parseur de données . . . 97
4.2.2.3 Le mouchard électronique EISEval . . . 98
4.2.3 Module 3 : Moteur de l'évaluation . . . 101
4.2.3.1 L'analyseur automatique des événements utilisateur . . . 101
4.2.3.2 L'analyseur de questionnaires . . . 106
4.2.3.3 Analyseur de l'interface graphique . . . 106
4.2.3.4 L'interpréteur des données . . . 107
4.2.4 Module 4 : Outil de génération de rapports . . . 109
4.3 Processus d'évaluation . . . 110
4.3.1 Phase de pré-évaluation . . . 110
4.3.2 Phase d'évaluation . . . 111
4.3.3 Phase de post-évaluation . . . 113
4.4 Discussion . . . 113

**5 Contribution 2 : Composants graphiques orientés évaluation des interfaces utilisateur 116**
5.1 Présentation générale de la contribution . . . 117
5.1.1 Motivation . . . 117
5.1.2 Concept . . . 118
5.2 Composants dédiés pour l'inspection de l'utilisabilité . . . 119
5.2.1 Modélisation et exploitation des règles ergonomiques . . . 120
5.2.2 Exploitation des règles ergonomiques au sein des composants pour l'évaluation . . . 123
5.3 Composants dédiés pour les tests utilisateur . . . 124
5.3.1 Mouchard Electronique . . . 125
5.3.1.1 Le générateur de modèles de référence . . . 126
5.3.1.2 Le générateur de modèles de la tâche observée . . . 127
5.3.1.3 Le module de confrontation . . . 128
5.3.1.4 Le module de génération de statistiques . . . 129
5.4 Processus d'évaluation adopté . . . 130
5.4.1 Processus de l'évaluation de l'utilisabilité à travers les composants orientés évaluation . . . 130
5.4.1.1 Phase de pré-évaluation/conception . . . 130
5.4.1.2 Phase d'évaluation/conception . . . 131
5.4.2 Processus de l'évaluation de l'utilité à travers les composants orientés évaluation . . . 131
5.4.2.1 Phase de pré-évaluation . . . 131

5.4.2.2 Phase d'évaluation . . . 132
5.4.2.3 Phase de post-évaluation . . . 133
5.5 Synthèse et discussion . . . 133

**III Mise en oeuvre 136**

**6 Evaluation expérimentale 137**
6.1 Première évaluation : évaluation du framework RITA . . . 138
6.1.1 Présentation du système à évaluer : SAI . . . 138
6.1.2 Processus d'évaluation . . . 142
6.1.2.1 Inspection de la cohérence des interfaces graphiques par rapport à des règles ergonomiques . . . 142
6.1.2.2 Capture de l'interaction avec des utilisateurs . . . 142
6.1.2.3 Questionnaire . . . 145
6.1.3 Résultats . . . 146
6.1.3.1 Inspection de la cohérence de l'interface graphique du SAI par rapport à des règles ergonomiques . . . 146
6.1.3.2 Interaction . . . 148
6.1.3.3 Questionnaire . . . 148
6.1.4 Synthèse des résultats et discussion . . . 149
6.2 Deuxième évaluation : évaluation des composants dédiés pour l'évaluation des interfaces utilisateur . . . 153
6.2.1 Présentation du système à évaluer : SAI . . . 153
6.2.2 Evaluation des composants dédiés pour l'inspection de l'utilisabilité des IU . . . 154
6.2.2.1 Conception/Evaluation . . . 154
6.2.2.2 Résultat de l'évaluation de l'IU de la nouvelle maquette du SAI . . . 155
6.2.3 Evaluation des composants dédiés pour l'inspection de l'utilité des *IU* . . . 155
6.2.3.1 Conception de l'interface . . . 156
6.2.3.2 Modèles de tâches . . . 156
6.2.3.3 Scénario d'expérimentation . . . 158
6.2.4 Processus d'évaluation . . . 158
6.2.4.1 Dispositif expérimental . . . 158
6.2.4.2 Résultats de l'évaluation . . . 158
6.3 Discussion et synthèse . . . 159

**Conclusion générale et Perspectives de recherche 163**

**IV Bibliographie** **169**

**V Annexes** **194**

**A Règles ergonomiques utilisées pour l'évaluation expérimentale** **196**

**B Questionnaire utilisé pour l'évaluation expérimentale** **198**

**C Synthèse sur des outils pour l'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques** **201**

**D Synthèse sur des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'interface et l'utilisateur** **205**

**Bibliographie** **210**

**Abstract** **210**

# Table des figures


1.1 Principe de l'évaluation des *IHM* suivant (Senach, 1990) . . . . . 6
1.2 Cycle d'évaluation des *IHM* (inspiré de (Ivory et Hearst, 2001)). . . . . 9
1.3 Classification de Senach (Senach, 1990). . . . . 13

2.1 Exemples positif et négatif de messages d'erreur. . . . . 32
2.2 Classification proposée des outils d'évaluation exploitant les connaissances ergonomiques. . . . . 37
2.3 Architecture de l'outil Ocawa . . . . . 42
2.4 Architecture de Sherlock (Inspiré de (Grammenos *et al.*, 2000)) . . . . . 44
2.5 Architecture de Destine (Inspiré de (Beirekdar *et al.*, 2002)) . . . . . 46
2.6 Google Analytics . . . . . 56

3.1 Extrait d'un exemple de décomposition d'une séquence d'action suivant l'arborescence proposée par Hilbert et Redmiles (Hilbert et Redmiles, 2000) 64
3.2 Exemple de résultats d'évaluation fournis par USINE (figure extraite de (Lecerof et Paterno, 1998)) . . . . . 67
3.3 Architecture fonctionnelle de RemUsine (Carta *et al.*, 2011) . . . . . 68
3.4 Exemple de résultats d'évaluation proposé par Wet . . . . . 69
3.5 Un exemple d'une capture de mouvement de la souris avec SMT (Speedzinemedia, 2012) . . . . . 71
3.6 Principe de couplage entre les agents du mouchard et les agents du système évalué (Trabelsi, 2006) . . . . . 73
3.7 Un exemple d'une heat map générée par ClikTale . . . . . 74
3.8 Exemple de heat map d'une page web générée par Click Heat (Labsmedia, 2012) . . . . . 76
3.9 Exemple d'interface de Morae . . . . . 77
3.10 Architecture générale d'EISEval (Tran *et al.*, 2008) . . . . . 78
3.11 Exemple d'image analysée par Feng-Gui . . . . . 80
3.12 Exemple d'une heat map issue de l'évaluation d'une page web avec Crazy Egg . . . . . 81
3.13 Un exemple de heat map généré par Usabilla los de l'évaluation d'un site web. . . . . 83
3.14 Exemple de résultat de capture de données avec Loop11 . . . . . 85
3.15 Silverback (Photo extraite de `www.Gigaom.com`). . . . . 86

4.1 Architecture du framework RITA . . . . . 94

4.2 Gestionnaire des règles ergonomiques : interface permettant l'ajout de nouvelles règles . . . 95
4.3 Gestionnaire des règles ergonomiques : interface permettant de spécifier une règle ergonomique . . . 96
4.4 Principe de génération de questionnaire dédié à l'évaluation des IHM . . . 97
4.5 Interface de saisie de questions . . . 98
4.6 Interface du parseur de données . . . 98
4.7 Exemple de décomposition d'une tâche d'EVIUs et d'EVDIs . . . 99
4.8 Aperçu de l'interface du mouchard électronique d'EISEval . . . 100
4.9 Aperçu de l'interface du sous-module « *Analyseur automatique des événements utilisateur* » . . . 102
4.10 Principe d'analyse automatique des actions utilisateur pour l'évaluation des IHM . . . 103
4.11 Exemple de RdP portant les probabilités de chaque transition . . . 104
4.12 Courbe illustrant le temps d'exécution et le nombre de tâche pour le calcul de la moyenne d'écart . . . 105
4.13 Principe de l'exploitation du questionnaire pour l'évaluation des IHM . . . 106
4.14 Principe de l'exploitation de l'inspection de la qualité ergonomique pour l'évaluation des IHM . . . 107
4.15 Interface de l'analyseur de l'interface graphique . . . 108
4.16 Interface de l'outil de génération de rapport d'évaluation . . . 110
4.17 Processus de pré-évaluation modélisé avec la notation BPMN . . . 111
4.18 Processus de l'évaluation modélisé avec la notation BPMN . . . 112

5.1 Exemple de composants orientés évaluation intégrés dans l'environnement « *MS Visual Studio 2010* » . . . 119
5.2 Capture écran de l'interface du « *The Ergonomic Guidelines Manager* » . . . 121
5.3 Processus de modélisation des règles ergonomiques sous forme de fichier XML. . . . 123
5.4 Architecture logicielle logique du mouchard électronique et des composants orientés évaluation . . . 125
5.5 Architecture fonctionnelle du mouchard électronique . . . 126
5.6 Interface du mouchard électronique dédié à l'évaluation des IU . . . 127
5.7 Exemple d'automate à états finis exploité pour l'évaluation des systèmes interactifs . . . 128
5.8 Exemple de résultats proposés par le mouchard électronique dédié à l'évaluation des IU . . . 130
5.9 Processus de l'évaluation à travers les composants dédiés pour les tests utilisateur . . . 132
5.10 Architecture fonctionnelle des composants orientés évaluation . . . 135

6.1 Architecture du système de supervision proposé dans le cadre du projet SART . . . 139
6.2 Interface du SAI : Interface « *Agent_Message* », Interface « *Agent_Etat_Ligne* » et Interface « *Agent_Etat_Trafic* » . . . 139
6.3 Architecture fonctionnelle du SAI . . . 140
6.4 Interface du SAD . . . 141
6.5 Interface du SAE . . . 142

6.6 Evolution du nombre de problèmes d'utilisabilité découverts en fonction du nombre d'utilisateurs (Nielsen et Landauer, 1993) . . . . . 143
6.7 Dispositif expérimental . . . . . 145
6.8 Une capture écran du questionnaire proposé pour les utilisateurs . . . . . 146
6.9 Courbe illustrant une synthèse sur les sessions d'interaction de dix utilisateurs avec le SAI. . . . . 148
6.10 Exemple de résultats du questionnaire . . . . . 149
6.11 Proposition d'une maquette d'interface du SAI . . . . . 152
6.12 Une maquette d'un nouveau prototype du SAI. . . . . 153
6.13 Une capture d'écran de la conception de l'interface utilisateur du SAI à l'aide des composants orientés inspection de l'utilité . . . . . 154
6.14 Modélisation des tâches requises pour l'utilisateur en CTT . . . . . 156
6.15 Capture d'écran de l'interface d'envoi de message avec le SAI . . . . . 157
6.16 Modélisation de la décomposition de la tâche utilisateur en sous-tâches en CTT . . . . . 157
6.17 Dispositif expérimental . . . . . 158
6.18 Aperçu d'un exemple de résultat de confrontation entre modèle de référence et modèle d'objet. . . . . 159

# Liste des tableaux

2.1 Les dix critères ergonomiques proposés par Mauney et Masterton pour les interfaces à petits écrans (Mauney et Masterton, 2008) . . . . . 33

4.1 Exemple de données pour l'évaluation des IHM . . . . . 108

6.1 Synthèse de résultat de l'inspection de la conformité de l'interface graphique du SAI par rapport à des règles ergonomiques . . . . . 147
6.2 Synthèse sur les sessions d'interaction de dix utilisateurs avec le SAI . . . 148
6.3 Association des règles ergonomiques et questions par rapport aux principes de la norme ISO 9241-111 (ISO, 1998) . . . . . 151
6.4 Exemples d'aspects considérés pour l'inspection de l'utilisabilité par les composants graphiques dédiés pour l'évaluation . . . . . 155

C.1 Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connsaisances ergonomiques (Partie 1) . . . . . 202
C.2 Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques (Partie 2) . . . . . 203
C.3 Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques (Partie 3) . . . . . 204

D.1 Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 1) . . 206
D.2 Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 2) . . 207
D.3 Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 3) . . 208
D.4 Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 4) . . 209

# Abréviations

| | |
|---|---|
| **AEF** | **A**utomate à **E**tats **F**inis |
| **BPMN** | **B**usiness **P**rocess **M**odeling **N**otation |
| **CTT** | **C**oncur **T**ask**T**rees |
| **CTTE** | **C**oncur **T**ask**T**rees **E**nvironment |
| **DLL** | **D**ynamic **L**ink **L**ibrary |
| **EARL** | **E**valuation **A**nd **R**eport **L**anguage |
| **EDI** | **E**nvironnement de **D**éveloppement **I**ntégré |
| **EVIU** | **EV**énement **I**nterface **U**tilisateur |
| **EVDI** | **EV**énement **D**ispositif **I**nteraction |
| **HTML** | **H**yper**T**ext **M**arkup **L**anguage |
| **IHM** | **I**nteraction **H**omme **M**achine |
| **IP** | **I**nternet **P**rotocol |
| **ISO** | **I**nternational **O**rganization for **S**tandardization |
| **IU** | **I**nterface **U**tilisateur |
| **PDA** | **P**ersonal **D**igital **A**ssistant |
| **RDL** | **R**eport **D**efinition **L**anguage |
| **RDP** | **R**eseau **d**e **P**etri |
| **RE** | **R**ègle **E**rgonomique |
| **RITA** | user**R**ègle **I**nterface evalua**T**ion fr**A**mework |
| **SI** | **S**ystème **I**nteractif |
| **SAD** | **S**ystème d'**A**ide à la **D**écision |
| **SAE** | **S**ystème d'**A**ide à l'**E**xploitation |
| **SAI** | **S**ystème d'**A**ide à l'**I**nformation voyageurs |
| **SOA** | **S**ervice **O**riented **A**rchitecture |
| **TIC** | **T**echnologies de l'**I**nformation et de la **C**ommunication |
| **UML** | **U**nified **M**odeling **L**anguage |
| **URL** | **U**niform **R**esource **L**ocator |
| **W3C** | **W**orld **W**ide **W**eb **C**onsortium |
| **WAI** | **W**eb **A**ccessibility **I**nitiative**C**onsortium |
| **WCAG** | **W**eb **C**ontent **A**ccessibility **G**uideline |
| **WIMP** | **W**indows, **I**cons, **M**enus, **P**ointer |
| **WUI** | **W**eb **U**ser **I**nterface |
| **WYSIWYG** | **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et |
| **XHTML** | e**X**tensible **H**yper**T**ext **M**arkup **L**anguage |
| **XML** | e**X**tensible **M**arkup **L**anguage |
| **XSLT** | e**X**tensible **S**tylesheet **L**anguage **T**ransformations |

*Je dédie ce mémoire à mes très chers parents Kamel et Hedia,*
*à mes grands parents que leurs âmes reposent en paix*
*et à mes chères soeurs Sana et Rawia.*

# Introduction générale

L'ère dans laquelle nous vivons est dite celle de l'information. En effet, le monde dans lequel nous vivons connait de jour en jour de grandes avancées au niveau des secteurs de télécommunications, de traitement automatique des informations, des technologies du web, de la nanotechnologie, etc. (Mattelart, 2006). Ces évolutions impliquent un accès de plus en plus fréquent, large et rapide aux différents systèmes permettant de communiquer les informations aux opérateurs humains. Ces systèmes sont dits interactifs du fait qu'ils assurent une interactivité avec leurs utilisateurs. Un des défis résultant de cette évolution exponentielle réside dans la qualité de la communication proposée entre ces systèmes et les opérateurs humains. Ce défi fait partie des objectifs de la communauté *IHM*. Cette dernière vise à promouvoir la communication entre utilisateur et systèmes interactifs via diverses techniques, méthodologies, concepts, etc. L'intérêt porté à la partie dédiée à la communication est accentué par le fait que cette partie est, elle seule, apparente à l'utilisateur (Meyer, 1997). De plus, les technologies ont atteint une évolution considérable que l'interface utilisateur est devenue un facteur de qualité (Bevan, 1999). Ce facteur peut être utilisé pour différencier entre les différents systèmes. Les travaux dans ce domaine de l'*IHM* sont nombreux et datent de plus d'une vingtaine d'années. Ces travaux portent essentiellement sur la conception, l'implémentation et l'évaluation des Interfaces Utilisateurs (*IU*) (Hewett, 1992).

Dans ce présent mémoire, nous nous focalisons principalement sur l'évaluation des *IU* comme étant un facteur primordial pour l'obtention de systèmes interactifs utiles et utilisables (Hearst, 2009). Les travaux relatifs à l'évaluation sont nombreux. Néanmoins, des difficultés liées à l'évaluation persistent encore. Cette thèse est une contribution à la thématique de l'évaluation des interfaces utilisateurs. Elle vise à mettre en place des outils pour l'évaluation de la partie interactive des systèmes interactifs. Elle s'articule essentiellement autour de deux contributions. La première porte sur un Framework dédié à l'évaluation des interfaces utilisateurs tout en se basant sur différentes techniques d'évaluation. La seconde contribution concerne des composants pour la composition

graphique d'interfaces utilisateur. Ces composants sont dotés de mécanismes supportant l'évaluation des *IU*. Ils sont répertoriés suivant deux familles. La première vise à assurer une auto-évaluation par rapport à des règles ergonomiques préalablement définies par l'évaluateur. La seconde, quant à elle, opère avec un mouchard électronique pour la capture et l'analyse de l'interaction entre l'utilisateur et le système interactif.

Cette thèse est structurée autour de trois parties. La première est consacrée à l'étude de l'existant. Elle s'étend sur les trois premiers chapitres. Le premier chapitre aborde la thématique de l'évaluation et les différentes techniques associées. Le second chapitre et le troisième chapitre s'intéressent à la présentation et l'étude des outils existants pour l'évaluation. Le second chapitre se restreint aux outils d'évaluation se basant sur des connaissances relatives à l'ergonomie logicielle pour l'évaluation des *IU*. Le troisième chapitre, quant à lui, s'adresse aux outils se basant sur la capture, l'analyse et l'étude de l'interaction entre le système interactif et l'utilisateur pour l'évaluation des *IU*. Cette partie vise à délimiter et à identifier les limites des outils existants pour l'évaluation des interfaces utilisateur. Ces limites nous amèneront à la seconde partie de ce mémoire : Contribution. Cette partie comporte deux chapitres. Ces chapitres présentent les deux contributions proposées lors de cette thèse. La première porte sur un Framework d'évaluation baptisé *RITA*. Ce dernier est présenté lors du quatrième chapitre. La seconde contribution concerne des composants pour la composition d'interfaces graphiques à l'aide d'Environnement de Développement Intégrés (*EDI* ) de type What You See Is What You Get (*WYSIWYG*). Ces composants ont la spécificité de supporter l'évaluation des *IU* depuis les premières phases de conception de systèmes interactifs. Ces composants sont répertoriés autour de deux familles. Ces deux familles visent à inspecter respectivement l'utilité et l'utilisabilité des interfaces utilisateur. La troisième partie du mémoire est dédiée à la mise en oeuvre des deux contributions proposées dans ce mémoire. Cette mise en oeuvre vise à valider les deux contributions et d'en déduire leurs limites. Ce mémoire se termine par une conclusion générale et des perspectives de recherche résultant des travaux élaborés lors de cette thèse.

Notons que cette thèse s'intègre dans le cadre du projet *CISIT*[1]-ISART[2] (2009-2013). Ce projet vise à promouvoir des systèmes de transport fiables et performants.

1. http://www.cisit.org/
2. Intégration des Systèmes d'Aide à la Régulation dans les Transports urbains multimodaux flexibles

# Première partie

# L'existant

Chapitre 1

# L'évaluation des systèmes interactifs vue sous l'angle de l'IHM

*« In general, technology's become so good that it's not the differentiator between products. User interface is becoming a huge differentiator. »*

Jefferson Han [1].

## Introduction

Depuis son apparition, l'informatique n'a cessé de croître à forte cadence. De ce fait, les systèmes interactifs sont devenus omniprésents dans tous les domaines et tous les secteurs (Karat et Karat, 2003). Ces systèmes ne cessent de devenir de plus en plus complexes et de moins en moins commodes à manier du fait qu'ils contiennent davantage d'informations et de fonctionnalités (Yvonne *et al.*, 2011). Ainsi, l'obtention de systèmes d'excellente qualité est devenue un défi croissant auquel la communauté du génie logiciel et de plus en plus celle de l'*IHM*[2] est souvent confrontée. Ce défi est accentué dans les domaines d'application sensibles aux erreurs de manipulation tel est le cas des domaines du transport, de la production d'énergie, de la santé, etc (Kortum, 2008; Johnson, 2000; Dixon *et al.*, 2001; Stephanidis, 2001). De ce fait, la communauté *IHM* (Interfaces Homme-Machine) a mené de nombreux travaux durant ces deux dernières

1. Jefferson Han est un chercheur à l'université de New York. Il s'agit d'un des majeurs auteurs du concept du multi-touch.

2. Le terme *IHM* a été adopté au milieu des années 80. Il existe plusieurs définitions de ce terme. L'ACM le définit comme étant une discipline qui s'intéresse à la conception, l'évaluation ainsi que l'implémentation de systèmes interactifs pour qu'ils soient utilités par des opérateurs humains tout en étudiant l'ensemble des phénomènes ayant lieu dans l'environnement de l'interaction (Hewett, 1992).

décennies. Cet intérêt s'est accentué, vu que l'*IHM* est, elle seule, la partie apparente à l'utilisateur (Meyer, 1997). Les centaines de milliers de lignes de code ne sont pas visibles à l'utilisateur. Beaucoup de travaux ont été élaborés dans ce contexte dans le but d'améliorer la partie servant d'interlocuteur entre le système et son utilisateur (Barkhuus et Rode, 2007; Dix, 2008). Ces travaux visent partiellement à promouvoir l'utilité (Abran *et al.*, 2003; ISO, 1999; Seffah *et al.*, 2006; Toomim *et al.*, 2011), l'utilisabilité (Nielsen, 1993; Stone *et al.*, 2005; Rubin et Chisnell, 2008; Blandford *et al.*, 2008; Lazar, 2007) et l'accessibilité (Thatcher *et al.*, 2006; Stephanidis *et al.*, 1997) des *IHM*. L'interface Homme-Machine est perçue même comme étant un facteur de qualité des systèmes interactifs (Law *et al.*, 2008; Fitzpatrick, 2003). De ce fait, sa validation et son test constituent une phase importante dans la conception de systèmes interactifs (Stone *et al.*, 2005).

Ce chapitre, intitulé « *Evaluation des systèmes interactifs vue sous l'angle des IHM* », en fait objet. Il est structuré autour de cinq sections. La première introduira le concept de l'évaluation des *IHM*. La seconde section présentera les classifications existantes des techniques[3] d'évaluation des *IHM*. Ensuite, la troisième section sera dédiée à la présentation d'une panoplie de techniques d'évaluation. La quatrième section présentera les difficultés liées à l'évaluation des *IHM*. Finalement, ce chapitre sera clôturé par une synthèse et une conclusion.

## 1.1 Concept de l'évaluation des *IHM*

L'évaluation des interfaces Homme-Machine consiste à tester et à valider les interfaces dans le but d'assurer un fonctionnement optimal des systèmes interactifs (Moussa *et al.*, 2006; Brangier, 2007). L'évaluation assure une augmentation de l'acceptabilité des systèmes auprès de l'utilisateur (Zhang *et al.*, 1999; Karapanos, 2005; Dillon et Morris, 1996). Il existe plusieurs définitions de l'évaluation des *IHM*. Dans certains travaux et outils, l'évaluation est définie comme étant une comparaison entre un modèle de référence et un modèle observé dans le but d'établir des conclusions (Senach, 1990; Kolski et Ezzedine, 2003). Par exemple, dans le cas des mouchards électroniques (Tran *et al.*, 2008), le processus d'évaluation consiste à comparer une suite d'actions élémentaires effectuées par l'utilisateur pour exécuter une tâche donnée (modèle observé) avec une

3. Dans plusieurs ressources, on utilise le terme méthode d'évaluation. Dans ce mémoire on opte plutôt le terme technique. En effet, suivant le dictionnaire Larousse, une méthode est un ensemble ordonné de manière logique de principes, de règles, d'étapes, qui constitue un moyen pour parvenir à un résultat. La technique quant à elle est définie comme étant un ensemble de procédés et de moyens pratiques propres à une activité. Ainsi, le terme technique nous semble plus adéquat du fait que lors des procédés d'évaluation il n'existe pas souvent de démarche bien ordonnée pour procéder à l'évaluation.

autre suite d'actions prévue par le concepteur du système pour l'exécution de cette tâche (modèle de référence), Figure 1.1.

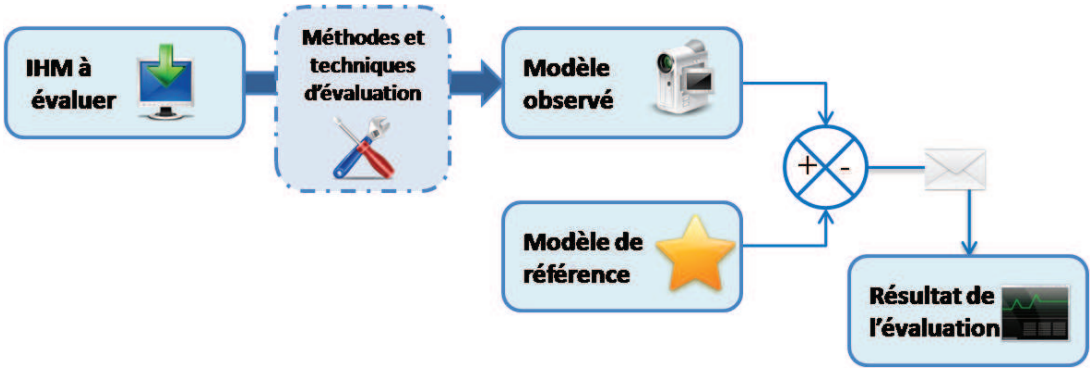



FIGURE 1.1: Principe de l'évaluation des *IHM* suivant (Senach, 1990)

D'autres travaux considèrent l'évaluation comme étant la mise en oeuvre sous la forme d'une validation de connaissances relatives à l'ergonomie logicielle, en lien avec le système interactif à évaluer *(Charfi *et al.*, 2011b; Bach et Scapin, 2010). Dans ce cas, elle consiste à assurer la conformité des interfaces utilisateur par rapport à des recommandations et des règles issues de l'ergonomie logicielle (Abascal *et al.*, 2004). A titre d'exemple, nous pouvons mentionner les techniques consistant à assurer le diagnostic de la qualité ergonomique des *IHM*, tel est le cas des guides ergonomiques (Chapitre 2). Une autre définition consiste à introduire l'évaluation comme étant la mesure de l'utilité et de l'utilisabilité du système évalué (Farenc *et al.*, 2001).

### 1.1.1 Objectifs de l'évaluation

Bien que les définitions de l'évaluation des *IHM* différent, elles convergent toutes vers les objectifs de l'évaluation. Ces dernières visent essentiellement à (Stephanidis, 2001) :

- détecter les aspects pouvant engendrer des problèmes d'utilisation : tel est le cas d'un problème lié à la lisibilité des informations présentées à l'utilisateur (Nielsen, 1993) ;
- améliorer l'acceptabilité des systèmes interactifs auprès des utilisateurs ; nombreux sont les recommandations pour la mise en place et la conception de systèmes interactifs. Ces recommandations sont généralement basées sur des études sur l'utilisateur (Zhang *et al.*, 1999) ;
- élaborer une comparaison entre plusieurs alternatives de conception (prototypes) (Mayhew, 1999) ;
- minimiser les risques d'erreurs (Wickens *et al.*, 2004) ;
- améliorer la sécurité des données de l'utilisateur et de leur manipulation en réduisant (voire même éliminer) les risques d'erreurs fatales (Olsen et Dan, 2007; Kuusela et Paul, 2000) ;

– mesurer la productivité et les performances du système (Rubin et Chisnell, 2008) ;
– protéger l'utilisateur des situations indésirables et des manipulations dangereuses du système (Shneiderman, 1992; Abowd *et al.*, 2004) ;
– comprendre le comportement et l'expérience de l'utilisateur (Wright *et al.*, 2006) dans le but de les exploiter dans la conception des systèmes interactifs (Seffah *et al.*) et,
– améliorer, voire même optimiser, l'efficacité et la productivité des systèmes interactifs (Zhang *et al.*, 1999).

Ainsi l'évaluation est une phase importante et essentielle pour garantir des systèmes interactifs de qualité.

### 1.1.2 Processus de l'évaluation

L'évaluation peut porter sur la présentation statique des éléments constituants les interfaces graphiques. Nous mentionnons le cas de l'agencement des éléments sur l'interface graphique. Elle peut aussi porter sur l'observation et l'étude des *IHM*. Dans la littérature il n'existe pas de standards formalisant le processus d'évaluation. Beaucoup de processus ont été proposés dans ce contexte. Le lecteur intéressé par les processus d'évaluation peut consulter (Seffah et Habieb-Mammar, 2009).

En s'inspirant des travaux de Ivory and Hearst (Ivory et Hearst, 2001), le processus de l'évaluation s'articule généralement autour de huit étapes essentielles, Figure. 1.2 :

1. **Définition des objectifs de l'évaluation.** Cette phase consiste à identifier les aspects et les critères à prendre en considération lors de l'évaluation ;
2. **Spécification du contexte.** A chaque utilisation d'un système interactif correspond un contexte d'usage. Le contexte est identifié par l'utilisateur (âge, compétences, performances, etc.), la plateforme d'interaction (capacité de calculs, dispositifs de calcul, etc.) et l'environnement physique (luminosité, mobilité, sonorisation, etc.) (Gabillon *et al.*, 2011) ;
3. **Mise en place du modèle d'évaluation.** Il s'agit de sélectionner les aspects et les critères à prendre en considération lors de l'évaluation. De plus, cette phase permet d'identifier clairement les objectifs de l'évaluation ;
4. **Choix de la technique de recueil des données pour l'évaluation.** Il existe diverses techniques pour le recueil des données d'évaluation (chaque technique capture des informations relatives à des aspects données). Il n'existe pas de technique idéale pour l'évaluation. Chacune de la panoplie des techniques est liée à un (des) contexte(s) donné(s) et a ses propres objectifs d'évaluation (Farenc *et al.*, 2001) ;
5. **Capture des données.** Cette phase permet de collecter les informations portant sur l'*IHM* , l'interaction entre l'utilisateur et le système et/ou l'avis de l'utilisateur

vis-à-vis du système. Les données collectées peuvent être à titre d'exemple les erreurs d'utilisation, temps d'exécution des tâches utilisateurs, l'avis des utilisateurs et les incohérences ergonomiques ;

6. **Analyse et traitement des données.** Une fois les données d'évaluation sont collectées, elles sont analysées et interprétées. Il existe différentes façons d'analyser. La plus utilisée est celle introduite par (Senach, 1990). Elle consiste à comparer deux modèles : le modèle de référence et le modèle observé. La comparaison vise à dégager et à identifier les problèmes éventuels dans l'*IHM* , Figure 1.3. Il existe d'autres mécanismes d'analyse inspirés du test logiciel tel est le cas du « *White Box Testing* »[4] (SIGIST, 2001; StandardSBoard, 1990) et le « *Black Box Testing* »[5] (Beizer, 1990; Copeland, 2004) ;
7. **Rapport.** Cette phase consiste à établir des rapports portants sur l'évaluation. Dans un premier temps le rapport émet un état concernant le déroulement du processus d'évaluation. Ensuite, une liste des différents problèmes détectés dans l'*IHM* évaluée est élaborée. Finalement, une liste de critiques concernant le système interactif évalué ainsi que des conseils pour l'amélioration du système est proposée à l'évaluateur afin de remédier aux différentes incohérences dans l'interface utilisateur et enfin ;
8. **Amélioration de l'interface.** Suite aux différents conseils et critiques de l'interface évaluée, le concepteur procède à l'amélioration de l'interface dans le but d'éliminer les aspects pouvant engendrer des problèmes d'utilisation.

Les quatre premières étapes peuvent être résumées en une seule étape. Cette étape consiste à planifier le processus d'évaluation via la réponse aux six questions suivantes :

1. Comment précéder à l'évaluation ? (**How**) : permet de sélectionner et choisir la technique appropriée pour l'évaluation ;
2. Quand évaluer ? (**When**) : permet de déterminer le stage de conception du système interactif dans lequel s'applique l'évaluation ;
3. Qui va procéder à l'évaluation ? (**Who**) : permet de déterminer les acteurs intervenants dans l'évaluation (utilisateurs, profil des évaluateurs, etc.) ;
4. Où procéder à l'évaluation (**Where**) : permet de spécifier le cadre spatial du processus d'évaluation ;

4. Le White Box Testing est une technique que les développeurs utilisent pour valider leurs codes sources. Cette technique prend en considération le mécanisme interne du système évalué. Cette technique est connue aussi sous le nom de test structurel. Son principal avantage est d'identifier la localisation des défaillances détectées (SIGIST, 2001; StandardSBoard, 1990).

5. Contrairement au White Box Testing, le Black Box Testing est une technique d'évaluation qui ne prend pas en considération la composition interne du système d'évaluation. Elle se base uniquement sur une comparaison entre les entrées et les sorties ou encore des sorties et les sorties prévues (Beizer, 1990; Copeland, 2004).

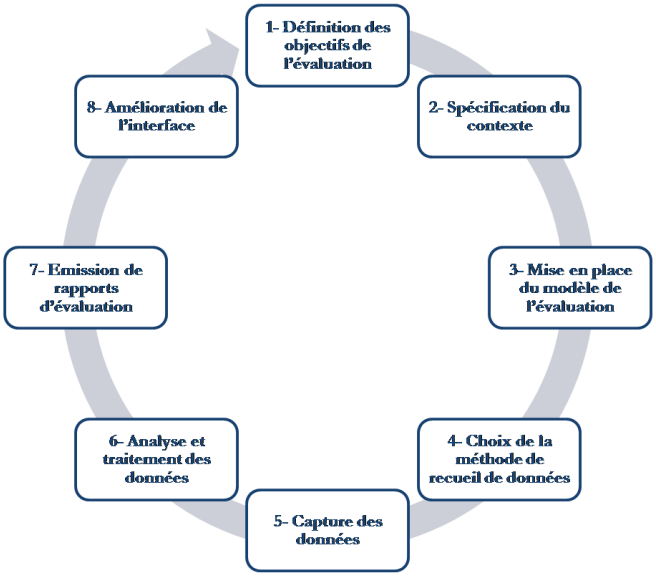



FIGURE 1.2: Cycle d'évaluation des *IHM* (inspiré de (Ivory et Hearst, 2001)).

5. Pourquoi évaluer ? (**Why**) : permet de renseigner sur l'objectif du processus d'évaluation et,
6. Qu'est-ce qu'on va évaluer et tester ? (**What**) : permet de mettre en place une liste des aspects du système interactif à évaluer.

## 1.2 Techniques d'évaluation

Les travaux sur les méthodes, outils et techniques d'évaluation des systèmes interactifs [6] remontent à plus d'une vingtaine d'années (Nielsen, 1993; Senach, 1990; Vanderdonckt, 1994). De ce fait les travaux dans ce contexte sont abondants. Ils s'articulent essentiellement autour des techniques pour l'évaluation (ou encore pour l'aide à l'évaluation) des systèmes interactifs. Les techniques proposées divergent selon un ensemble de points.

1. **Nature des données pour l'évaluation.**
   (a) Données quantitatives. Il s'agit généralement de données numériques. Ces données sont généralement des métriques portant sur les performances du système interactif évalué. L'évaluation se basant sur de telle évaluation peut renseigner l'évaluateur sur le taux d'erreur pour l'exécution de certaines tâches, les fonctions et les composants inutilisés, la durée d'apprentissage, la durée d'exécution d'une tâche, etc. Dans ce contexte, nous nous référons à l'exemple du mouchard électronique (Tran *et al.*, 2008).

6. Dans ce mémoire, le terme évaluation des systèmes interactifs exprime l'évaluation des interfaces utilisateurs des systèmes interactifs.

(b) Données qualitatives. Ces données ne sont pas numériques. Elles sont élaborées via la collecte des opinions des utilisateurs suite à des entretiens (Olsen, 2002; Kuusela et Paul, 2000), à des observations (Holzinger, 2004; Bentley *et al.*, 1992) ou encore à des questionnaires (Bark et Gulliksen, 2006). Elles servent à établir une liste de problèmes liés au système évalué. De plus, elles permettent d'interroger l'utilisateur sur l'aspect esthétique du système et ses préférences. A titre d'exemple, nous citons l'exemple de l'observation des utilisateurs comme technique d'évaluation des *IHM* (Holzinger, 2004; Bentley *et al.*, 1992).

2. **L'élément de qualité logicielle pris en considération lors de l'évaluation.** Généralement les techniques d'évaluation se basent sur l'utilisabilité (Nielsen, 1993; Jarrett et Gaffney, 2009; Schaffer, 2004; Stone *et al.*, 2005; Rubin et Chisnell, 2008; Blandford *et al.*, 2008; Lazar, 2007), l'utilité (ISO, 1999; Abran *et al.*, 2003; Seffah *et al.*, 2006; Toomim *et al.*, 2011), et/ou l'accessibilité (Thatcher *et al.*, 2006; Stephanidis *et al.*, 1997). L'utilisabilité est définie par la norme *ISO/IEC 9241* comme étant : l'efficacité, l'efficience et la satisfaction avec laquelle les utilisateurs atteignent les objectifs spécifiques dans des environnements particuliers (ISO, 1999; Stone *et al.*, 2005; Rubin et Chisnell, 2008; Lazar, 2007; Osterbauer *et al.*, 2010). L'utilisabilité est considérée comme étant un facteur primordial et indispensable dans les *IHM*. Certains associent étroitement l'utilisabilité avec l'*IHM* tel est le cas de Caroll en définissant l'*IHM* par l'étude et la pratique de l'utilisabilité (Carroll, 2002; Jarrett et Gaffney, 2009). L'utilité d'un système détermine si ce dernier permet à l'utilisateur de réaliser sa tâche. Autrement dit, elle répond à la question suivante : « **L'utilisateur est-il capable de réaliser ce qui est nécessaire pour satisfaire les besoins pour lesquels le système a été créé ?** ». L'utilité correspond, donc, aux capacités fonctionnelles, aux performances du système et à la qualité de l'assistance technique fournie à l'utilisateur par le système (Abran *et al.*, 2003; Seffah *et al.*, 2006). L'accessibilité d'une *IHM* est définie comme étant sa capacité à être utilisée et exploitée par un public le plus large possible. Il est bien difficile de mettre au point des applications accessibles par tout type d'utilisateurs du fait qu'il existe beaucoup de différences entre eux : culturelles, linguistiques, sociales, techniques, niveau d'expertise à manier les systèmes interactifs, etc. (Thatcher *et al.*, 2006; Stephanidis *et al.*, 1997).

3. **Phase d'application dans le cycle de conception des systèmes interactifs.**
Les techniques d'évaluation peuvent diverger à cause du moment d'application. En effet, certaines s'appliquent lors des premières phases du cycle de conception, il s'agit du cas de l'évaluation précoce (Tarby *et al.*, 2009). Cette dernière consiste à

coupler la conception explicitement avec l'évaluation (Nielsen, 1993; Tarby *et al.*, 2009; Ezzedine *et al.*, 2006). L'avantage de cette évaluation précoce est de permettre un gain de temps important lors de l'évaluation de la partie interactive des systèmes. En effet, l'évaluateur est dispensé de procéder à des améliorations, des modifications et des corrections du système interactif déjà implémenté. De plus, les corrections peuvent s'avérer coûteuses (en termes d'effort et de temps). Suivant *Nielsen*, il peut revenir 100 fois plus cher d'établir des corrections du système déjà réalisé que de le corriger au début du cycle de développement (Nielsen, 1993). D'autres techniques ne s'appliquent que durant la phase de spécification. Elles s'adressent aux systèmes interactifs non encore finis dans le but de l'améliorer (du point de vue *IHM*). Une troisième collection regroupe les techniques utilisables après la mise en exploitation des systèmes interactifs. Comme exemple, nous pouvons citer les tests et les enquêtes auprès des utilisateurs des systèmes interactifs.

4. **Degrés de l'automatisation du processus d'évaluation proposé.**
   L'automatisation vise à promouvoir la subjectivité du processus d'évaluation. Ainsi les résultats sont plus fiables. Par exemple, Balbo distingue essentiellement quatre catégories de processus d'évaluation (Balbo, 1996).

   (a) Non automatique. L'ensemble des techniques exécutées par des opérateurs humains.

   (b) Capture automatique. Il s'agit des techniques recourant à des outils généralement logiciels pour l'enregistrement et la capture des informations relatives à l'interaction entre le système évalué et l'utilisateur ;

   (c) Analyse automatique. Regroupe les techniques identifiant automatiquement les erreurs de conception dans le système évalué et,

   (d) Critique automatique. Il s'agit des techniques qui proposent, en plus de l'analyse automatique, des suggestions d'amélioration du système interactif.

5. **La nature des problèmes détectés dans une IHM.**
   Les problèmes peuvent être d'ordre (Nielsen, 1993) :

   (a) Critique. (ou encore catastrophique) Il s'agit des erreurs dont leur correction est impérative avant d'utiliser le système (que ce soit pour l'exploiter ou bien pour le tester),

   (b) Majeur. Il s'agit de l'ensemble d'erreurs qui sont prioritaires au niveau de l'amélioration du système évalué,

   (c) Moyen. Ce sont les erreurs d'importance moyennes. Ils n'engendrent pas de grands problèmes au niveau de l'utilisation mais il n'est pas recommandé de les ignorer,

(d) Mineur. Ce sont les erreurs dont l'évaluateur n'est pas amené à corriger mais, néanmoins, leur correction est préférable et,

(e) Cosmétique. Il s'agit de l'ensemble des erreurs qui ne sont pas importantes et que leur correction n'est pas nécessaire dans le cas d'insuffisance de ressources ou bien de temps.

Dans ce qui suit nous allons aborder les classifications les plus utilisées et référencées dans la littérature.

### 1.2.1 Classifications existantes

Il existe de nombreuses classifications de techniques d'évaluation des *IHM*. Nous citons notamment (en ordre chronologique d'apparition) :

1. **La classification de Howard (Howard et Murray, 1987).** Elle consiste à distinguer deux classes d'évaluation. La première est intitulée « *évaluation formative* ». Elle est effectuée lors de la phase de conception. La seconde, quant à elle, est intitulée « *évaluation sommative* ». Les techniques relatives à cette classe sont destinées à l'usage avec le système fini.
2. **La classification de Senach (Senach, 1990).** Senach distingue deux catégories essentielles de techniques d'évaluation : approches analytiques (expérimentales) et approches empiriques (prédictives). Néanmoins, il propose des sous catégories. En effet, les approches empiriques sont constituées de test de diagnostic d'usage et de test de conception. Les approches analytiques sont répertoriées en approches formelles (expertise et check-list) et informelles (modèles prédictifs et modèles de qualité) ; Figure 1.3.

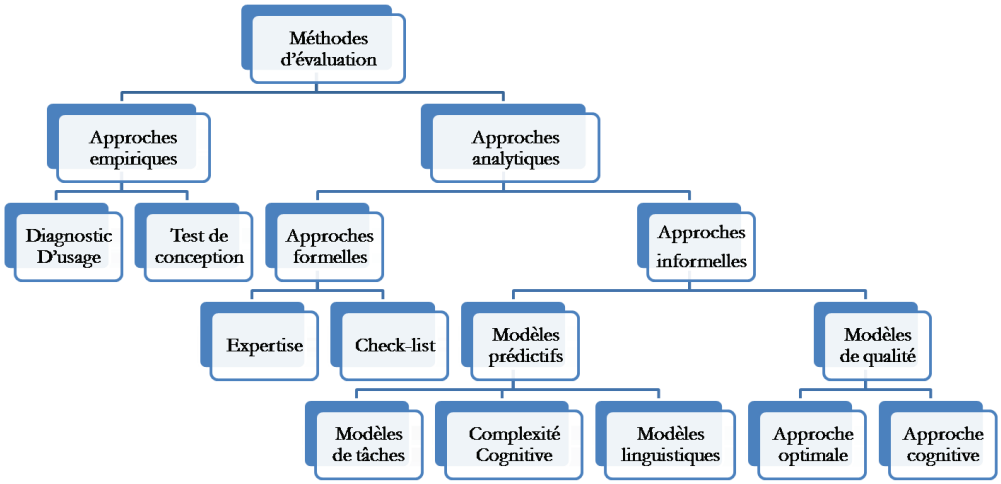



FIGURE 1.3: Classification de Senach (Senach, 1990).

3. **La classification de Coutaz (Coutaz, 1990).** Selon cette classification, les techniques d'évaluation s'articulent autour de deux catégories. La première s'appuie sur la représentation théorique du système évalué ; il s'agit de l'évaluation « *prédictive* ». La seconde porte sur une représentation du système (prototype, système final, etc.) : « *l'évaluation expérimentale* ».

4. **La classification de Balbo (Balbo, 1996).** Cette classification se base sur un ensemble de facteurs. Il s'agit des données requises, les ressources matérielles [7] nécessaires, les facteurs situationnelles [8], les moyens humains et le résultat fourni pour l'évaluation.

5. **La classification de Grislin et Kolski (Grislin et Kolski, 1996).** Cette classification comporte trois grandes catégories. La première concerne les approches centrées utilisateur. La seconde comporte les approches centrées sur les experts. La troisième catégorie est relative aux approches centrées sur une modélisation.

6. **La classification de Farenc (Farenc, 1997).** Cette classification se base sur trois éléments de l'interaction Homme-Machine : l'utilisateur, la tâche et le système.

7. **La classification de Bastien et Scapin (Bastien et Scapin, 2001).** Bastien et Scapin distinguent deux catégories de techniques. Leur classification se base sur

7. Les ressources matérielles sont répertoriées en trois catégories. La première regroupe les périphériques de capture de données pour l'évaluation. La seconde concerne l'objet de l'évaluation (maquette, prototype avancé, produit final, etc.). Et la troisième comporte le niveau d'abstraction des données capturées.

8. Les facteurs situationnels comportent le contexte d'évaluation. Ce contexte est constitué essentiellement de cinq éléments : la phase du cycle de développement, le lieu de l'évaluation, le type du système évalué, la nature de l'interface évaluée et le planning temporel du processus d'évaluation.

la participation des utilisateurs dans le processus d'évaluation comme un critère de classification. De ce fait, la première catégorie regroupe les techniques requérant la participation des utilisateurs tel est le cas des mouchards électroniques et de l'observation de l'utilisateur (Holzinger, 2004). La seconde, quant à elle, concerne les techniques ne requérant pas la participation de l'utilisateur.

8. **La classification d'Ivory et Hearst (Ivory et Hearst, 2001).** Les auteurs répertorient les techniques d'évaluation autour de cinq catégories :

   (a) Enquête. Elle consiste à établir un état portant sur les éléments que les utilisateurs apprécient dans le système, les éléments qu'ils n'apprécient pas ainsi que la vision du système par l'utilisateur. Le recueil des données est généralement effectué via des interviews (Olsen, 2002; Kuusela et Paul, 2000), en les observant en interaction avec le système dans un environnent de travail réel ou encore avec des questionnaires (oral ou écrit) (Bark et Gulliksen, 2006). Exemple de techniques : Observations (Holzinger, 2004; Bentley *et al.*, 1992), interviews (Olsen, 2002; Kuusela et Paul, 2000), questionnaires (McNamara et Kirakowski, 2011; Bark et Gulliksen, 2006; Kovacs *et al.*, 2004), enquête contextuelles (Beyer et Holtzblatt, 1998), focus groups (Nielsen, 1997), sondage, etc.

   (b) Inspection de l'utilisabilité des systèmes interactifs. Elle est effectuée généralement par des experts et des spécialistes en utilisabilité. L'utilisabilité des systèmes interactifs peut être inspectée par les concepteurs de systèmes interactifs (Jarrett et Gaffney, 2009). Elle consiste essentiellement à inspecter et à tester les aspects relatifs à l'utilisabilité du système évalué (Schaffer, 2004). A titre d'exemple, nous citons comme techniques : l'évaluation heuristique (Nielsen et Molich, 1990), la *Cognitive Walkthrough* (Mahatody *et al.*, 2010), les tests formels d'utilisabilité, l'inspection de la cohérence, les normes (telles que les normes ISO/IEC 9126 et les normes AFNOR Z67-110) et les guides ergonomiques (cf. Chapitre 2).

   (c) Test utilisateur. Le test du système interactif consiste à ce que des utilisateurs procèdent à l'exécution de certaines tâches avec le système évalué. Les utilisateurs évoqués lors de ce test peuvent être les utilisateurs finaux du système ou encore un échantillon représentatif des utilisateurs potentiels du système. Ceci a pour but d'identifier la façon avec laquelle le système répond aux requêtes des utilisateurs. Les techniques associées à cette catégorie visent aussi à collecter les données concernant l'interaction assurée avec l'utilisateur. Parmi les techniques relatives à cette catégorie, nous citons : le protocole *Think Aloud* (Jaaskelainen, 2000; Olmsted-Hawala *et al.*, 2010), l'oculométrie (Beckers et

Fuhr, 2010), le mouchard électronique (Tran *et al.*, 2008) et la mesure des performances des systèmes (Jeffries *et al.*, 1991).

(d) Les techniques analytiques et prédictives. Cette catégorie de techniques consiste à prédire la réaction de l'utilisateur vis-à-vis du système interactif. Elle vise à simplifier l'interaction entre l'utilisateur et le système. Généralement les techniques relatives à cette catégorie, s'appliquent lors des premières phases de conception des systèmes interactifs. Comme exemple, nous citons *GOMS*[9] (Schrepp et Fischer, 2007) et l'analyse des tâches (Crandall *et al.*, 2006; Crystal et Ellington, 2004).

(e) Simulation. Cette catégorie consiste à simuler le comportement de l'utilisateur vis-à- vis du système évalué par l'évaluateur. Ensuite, l'évaluation porte sur l'analyse et l'exploitation des données issues de cette simulation telle que le déroulement des actions exécutées, le temps d'exécution et les erreurs détectées lors de l'interaction.

### 1.2.2 Proposition de classification des techniques d'évaluation

Nous avons vu qu'il existe de nombreuses classifications des techniques d'évaluation. Chacune d'entre elle se base sur un ou plusieurs facteurs pour répertorier les techniques. Il se trouve que ces classifications ne facilitent pas vraiment la tâche de l'évaluateur pour sélectionner une technique appropriée pour le processus d'évaluation. Nous proposons dans cette sous-section de répertorier les techniques suivant les facteurs qu'elles inspectent. Nous distinguons essentiellement trois facteurs : l'utilisabilité, l'utilité et l'accessibilité.

1. **Techniques d'inspection de l'utilisabilité des *IHM*.** ces techniques ne requièrent pas nécessairement la présence de l'utilisateur. Généralement, ils procèdent à des tests de l'interface du système interactif pour dégager les problèmes d'utilisabilité (Jarrett et Gaffney, 2009). Ces problèmes portent souvent sur la présentation statique des informations sur l'interface utilisateur. Dans cette catégorie, il n'est pas obligatoire de disposer du système interactif fini ou encore d'un prototype avancé (Carter et Hundhausen, 2010) ; une maquette du système interactif suffit pour procéder à l'évaluation.
2. **Techniques d'inspection de l'utilité des *IHM*.** Ces techniques s'intéressent plutôt à l'interaction entre l'utilisateur et le système. Ils prennent en considération le noyau fonctionnel du système. L'utilité recouvre trois niveaux de question : « ***comment***, ***quoi*** *et* ***pourquoi*** *?* ». A rappeler que l'utilité consiste à mesurer à

9. GOMS est un acronyme pour **G**oals, **O**perators, **M**ethods and **S**election rules. Il permet de modéliser les différents niveaux du comportement utilisateur.

quel point le système permet à l'utilisateur d'atteindre les objectifs pour lesquels le système a été conçu (Toomim *et al.*, 2011). Il s'agit d'un facteur essentiel et primordial. En effet, un système peut être facile à manier et à comprendre mais il n'est pas conforme aux objectifs spécifiques de l'utilisateur. Malheureusement, il est souvent négligé lors du processus d'évaluation (Rubin et Chisnell, 2008).

3. **Techniques d'inspection de l'accessibilité des *IHM***. Généralement elles sont adressées à l'évaluation des pages web afin d'offrir une grande audience aux sites web. En effet l'accessibilité est une notion assez proche de la facilité d'utilisation du système interactif. Rubin la définit comme étant l'accès aux données nécessaires pour accomplir un objectif donné (Rubin et Chisnell, 2008). Les techniques relevant de cette catégorie ont pour objectif de garantir que le système évalué peut être manié et utilisé par la population la plus large possible d'utilisateurs, y compris ceux qui ont un handicap ou bien interagissant dans un contexte spécifique. Cette catégorie est bien riche en techniques et recommandations (Stephanidis *et al.*, 1997). Ces recommandations proviennent essentiellement des directives établies par le W3C [10]. Ces directives sont intitulées WAI [11].

## 1.3 Les techniques d'évaluation

Cette section propose un aperçu des techniques d'évaluation des *IHM*. Les techniques sont si variées et nombreuses, qu'on se propose lors de cette section de faire une synthèse des techniques les plus utilisées et les plus pertinentes.

### 1.3.1 Classification adoptée

Dans la section précédente, nous avons vu qu'il existe plusieurs classifications des techniques d'évaluation. Dans cette section, nous allons adopter celle proposée par Ivory et Hearst (Ivory et Hearst, 2001). Nous allons nous focaliser sur uniquement trois catégories de techniques : Enquête, inspection de l'utilisabilité des systèmes interactifs et test utilisateur. Ce choix est du au fait que les techniques analytiques et prédictives sont plutôt focalisées conception qu'évaluation. Quant aux techniques relatives à la catégorie de simulation, nous estimons qu'il est inutile de détailler les techniques associées du fait que nous allons nous focaliser sur les techniques de test utilisateur.

10. **W**orld **W**ide **W**eb **C**onsortium : est un organisme de normalisation qui vise à promouvoir la compatibilité des technologies du web. http://www.w3.org/

11. Web Accessibility Initiatives : il s'agit d'un ensemble de directive visant à rendre les différentes sites web accessibles aux personnes ayant des handicapes. http://www.w3.org/WAI/

### 1.3.2 Format

Les techniques présentées dans cette section seront introduites suivant un format :
- Objectifs : les objectifs visés par la technique d'évaluation ;
- Description : La description du protocole proposé pour l'évaluation des *IHM* ;
- Avantages : les avantages de la technique par rapport aux techniques d'évaluation existantes ;
- Inconvénients : les inconvénients de l'utilisation de la technique proposée ;
- Intervenants : les acteurs intervenants lors du processus d'évaluation ;
- Ressources matérielles requises : le dispositif matériel requis pour assurer le processus d'évaluation et,
- Positionnement dans le cycle de conception des systèmes interactifs : dans cet aspect, nous allons nous baser sur le cycle de développement en cascade (Larman et Basili, 2003; Medvidovic et Jakobac, 2005).

### 1.3.3 Panorama des techniques d'évaluation

#### 1.3.3.1 Techniques relevant de la classe d'enquête

**Observation (Holzinger, 2004; Bentley *et al.*, 1992)**
Objectif : Observer l'utilisateur interagissant avec le système interactif à évaluer en but d'analyser son comportement.

Description : Observer les utilisateurs lors de l'interaction avec le système évalué dans le but de détecter les aspects pouvant engendrer des problèmes d'utilisation. L'observation peut s'effectuer grâce à des enregistrements vidéo, audio ou encore en observation directe par un ou plusieurs évaluateurs. L'observation peut s'effectuer dans un environnement réel du système ou encore dans un environnement similaire (Bentley *et al.*, 1992; Holzinger, 2004).

Avantages :
- une technique facile à mettre en place et ne requérant pas de connaissances spécifiques pour l'évaluation ;
- ne nécessite pas un nombre important d'intervenant dans le processus d'évaluation ;
- son efficacité pour détecter des problèmes d'utilité et d'utilisabilité ;
- implique l'utilisateur final dans l'évaluation (Farenc, 1997) et,
- prend en considération le contexte d'utilisation dans l'évaluation du système (Rauterberg, 1996).

Inconvénients :
- la difficulté du choix de la population d'utilisateurs pour l'observation (nombre, profil, expérience, etc.) ;
- la difficulté de l'observation et de l'analyse de l'interaction entre l'utilisateur et le système ;
- repose sur des jugements d'ordre subjectif qui varient d'un évaluateur à un autre (Jeffries *et al.*, 1991; Molich *et al.*, 1999) et,
- cette technique est un peu gourmande en matière de temps et de ressources (Farenc *et al.*, 2001).

Intervenants : l'évaluateur, le développeur (facultatif) et un groupe d'utilisateurs.

Ressources matérielles : des périphériques d'enregistrement comme une caméra (facultatif) et le système à évaluer l'*IHM*.

Phase d'application dans le cycle de développement : Cette technique peut être appliquée depuis les premières phases du cycle de développement en exploitant une maquette du système ou encore un prototype.

**Interview (Olsen, 2002; Kuusela et Paul, 2000)**

Objectif : Interroger l'utilisateur(s) sur les éventuels problèmes d'utilité et d'utilisabilité qu'il a vus dans l'interface. Elle peut indiquer sur certaines dimensions telles que la facilité d'utilisation, le guidage, etc.

Description : L'interview consiste à s'entretenir avec des utilisateurs du système conçu dans le but de dégager auprès d'eux les problèmes qu'ils ont eu en interagissant avec le système interactif. L'évaluateur peut aussi demander aux utilisateurs de justifier certains de leurs choix ou encore de leur expliquer le déroulement d'actions adopté par eux. Géographiquement, l'interview peut avoir lieu sur le site d'utilisation du système évalué ou ailleurs (Olsen, 2002).

Avantages :
- cette technique ne requiert pas de grands moyens matériels pour sa mise en place ;
- la flexibilité de cette technique ; l'évaluateur dispose d'une liberté pour diriger les questions orales et les orienter sur les aspects qu'il désire (Trabelsi, 2006) et,
- permet à l'utilisateur de s'exprimer clairement concernant son avis sur la qualité de dialogue avec le système interactif.

Inconvénients :

- Pareil que l'observation, nous retrouvons une difficulté au niveau du choix de la population d'utilisateurs pour l'observation (nombre, profil, etc.) ;

– la difficulté de l'observation et de l'analyse de l'interaction entre l'utilisateur et le système ;
– repose sur des jugements d'ordre subjectif qui varient d'un évaluateur à un autre et,
– l'impossibilité d'automatiser l'entretien en vue d'évaluer le système interactif.

Intervenants : l'évaluateur et un ou plusieurs utilisateurs.

Ressources matérielles : aucune ressource n'est requise pour cette technique.

Phase d'application dans le cycle de développement : phase finale (phase de test).

**Questionnaire (Bark et Gulliksen, 2006)**

Objectif : Cette technique permet aux évaluateurs de recueillir un ensemble d'appréciations, d'attitude et d'opinions des utilisateurs vis-à-vis du système. L'évaluateur peut aussi demander aux utilisateurs de justifier certains de leur choix ou encore de leurs expliquer le déroulement d'actions adopté par l'utilisateur (Nielsen, 1993; McNamara et Kirakowski, 2011).

Description : L'évaluation d'une *IHM* à l'aide du questionnaire consiste à proposer un ensemble de questions bien structurées (sur papier ou encore sur le web) à un ensemble d'utilisateurs du système. Les réponses des utilisateurs sont par la suite analysées et interprétées pour renseigner sur les problèmes de satisfaction des utilisateurs. Le questionnaire proposé doit être bien établi avec des finalités bien claires et nettes dans le but d'éviter la redondance et les questions inutiles (Kovacs *et al.*, 2004). Comme exemples illustratifs, nous pouvons citer : SUMI [12] et SUS [13].

Avantages :
– technique intéressante en termes d'absence de ressources matérielles et en temps d'exécution ;
– sa facilité d'utilisation et,
– permet d'extraire des données pour l'évaluation qu'il n'est pas possible de collecter autrement.

Inconvénients :
– difficulté des choix et de l'élaboration des questions et,
– difficulté d'analyser les réponses des utilisateurs pour dégager l'ensemble des corrections à établir dans le système interactif.

12. Software Usability Measurement Inventory un outil dédié pour la mesure de la qualité des systèmes logiciels du point de vue utilisateur. Il peut être appliqué pour un prototype de logiciel afin de détecter les erreurs liées à l'utilisabilité dans ces systèmes logiciels. http://sumi.ucc.ie

13. System Usability Scale est un questionnaire composé par dix questions afin de recueillir l'appréciation des utilisateurs vis-à-vis d'un système donné. Les réponses sont recueillies suivant l'échelle de Likert allant du pas du tout d'accord au parfaitement d'accord. Il s'agit d'un questionnaire rapide à mettre en place (Brooke, 1996).

Intervenants : l'évaluateur, le développeur (facultatif) et un groupe d'utilisateurs.

Ressources matérielles : généralement le questionnaire se fait sur papier ou via un site web.

Phase d'application dans le cycle de développement : phase de conception (pour recueillir les attentes de l'utilisateur) et phase de test (pour recueillir l'avis de l'utilisateur).

#### 1.3.3.2 Techniques relevant de la classe d'inspection de l'utilisabilité

**Inspection de la qualité ergonomique** (Nielsen et Molich, 1990)
Objectif : Dégager les problèmes relatifs à la conception de l'interface du système interactif.
Description : Consiste à inspecter la qualité ergonomique d'une interface utilisateur selon des principes bien déterminés. Parmi ces principes, nous pouvons citer les heuristiques proposées par Nielsen [14] (Nielsen, 1993; Nielsen et Molich, 1990), les critères de Bastien et Scapin (Bastien et Scapin, 1993) ou encore les normes ISO (ISO, 1998).

Avantages :

– Il s'agit d'une technique rapide, non coûteuse et facile pour l'évaluation de la conception des *IU*.

Inconvénients :

14. Les dix principes de Nielsen sont (Nielsen, 1993; Nielsen et Molich, 1990) :

- la visibilité de l'état du système : les utilisateurs doivent être informés de ce que le système est en train de faire ;
- la corrélation entre le système et le monde réel : le système doit adopter le langage et les concepts de l'utilisateur pour assurer une meilleure communication ;
- le contrôle sur le processus : l'utilisateur doit pouvoir exécuter ses tâches sans procéder à des dialogues étendus avec le système ;
- conformité aux standards : le système doit être conforme aux standards pour afficher les informations ;
- prévention contre les erreurs : prévenir l'utilisateur des erreurs de manipulation via des messages d'erreurs bien claires afin de minimiser les risques d'erreurs ;
- reconnaissance : réduire la charge en mémoire de l'utilisateur via l'utilisation d'objets faisant référence aux informations de l'interface ;
- flexibilité et efficacité : le système doit répondre à la fois aux utilisateurs novices et ceux expérimentés en leur autorisation à adapter les actions fréquentes ;
- une conception esthétique et minimaliste de l'interface : l'interface ne doit pas comporter des informations inutiles ou bien rarement nécessaires ;
- aider les utilisateurs à reconnaitre, diagnostiquer et corriger les erreurs : les messages d'erreurs doivent être exprimés en un langage simple tout en indiquant le problème et proposer des solutions constructives et,
- aide et documentation : il est bien meilleur de munir le système avec une aide et une documentation qu'il soit plus facile à manier par l'utilisateur.

– Nielsen (l'auteur des heuristiques) énonce que l'évaluation heuristique ne peut détecter qu'en moyenne 35% des problèmes d'utilisabilité dans l'interface utilisateur d'un système interactif par un seul évaluateur (Nielsen, 1993) et,
– ne permet pas de résoudre les problèmes d'utilisabilité, elle permet seulement leur détection.

Intervenants : Nielsen recommande le recours à plus d'un évaluateur ayant l'habitude à évaluer les *IHM* selon ses dix principes (Nielsen, 1993).

Ressources matérielles : aucune ressource matérielle n'est requise pour l'évaluation.

Phase d'application dans le cycle de développement : peut être appliquée à partir de la fin de la phase de conception de l'interface graphique : phase de test.

**Cognitive Walkthrough (Mahatody *et al.*, 2010)**

Objectif : Prédire les éventuels problèmes d'utilisabilité dans un système interactif. Elle concerne l'inspection de la facilité d'utilisation et d'apprentissage du système évalué (Mahatody *et al.*, 2010).

Description : La *Cognitive Walkthrough* consiste à simuler le comportement d'un utilisateur d'un système interactif en exécutant des tâches avec le système. Cette simulation permet de prédire les problèmes que peut rencontrer l'utilisateur. Généralement l'évaluateur est aidé par un ensemble de questions dans le but d'évaluer le maximum d'aspects dans un système interactif (Polson *et al.*, 1992).

Avantages :
– Le faible coût du processus d'évaluation proposé (temps et ressources).

Inconvénients :
– l'évaluateur ne peut pas vraiment reproduire le comportement utilisateur avec l'*IHM* évaluée et,
– selon une étude menée par Lewis (Lewis *et al.*, 1990), la *Cognitive Walkthrough* ne peut pas couvrir plus que la moitié des problèmes d'utilisabilité dans un système interactif.

Intervenants : l'évaluateur.

Ressources matérielles : La Cognitive Walkthrough ne nécessite que le système interactif pour procéder à l'évaluation.

Phase d'application dans le cycle de développement : peut être appliquée à partir de la fin de la phase de conception de système interactif : phase de test.

### Guides ergonomiques [15]

Objectif : Détecter les problèmes de présentation des informations sur une interface utilisateur (Almeida et Baranauskas, 2010; Vanderdonckt, 1994).

Description : Il existe plusieurs règles ergonomiques dans la littérature (Vanderdonckt, 1994). Ces règles ont pour objectif de guider le concepteur et/ou l'évaluateur à obtenir des interfaces utilisateur ergonomiques acceptables par les utilisateurs finaux. Le principe d'évaluation est assez simple. Il consiste à inspecter la conformité de l'interface graphique par rapport à un ensemble des règles ergonomiques présélectionnées.

Avantages :
- simplicité du processus d'évaluation proposé et,
- les guides ergonomiques comprennent des règles qui sont exprimées en langage naturel et indépendamment de tout contexte d'utilisation (Keith, 2005; Rauterberg, 1996) ;

Inconvénients :
- le grand choix des règles ergonomiques. En effet, les règles existants pour l'inspection de la qualité ergonomique des interfaces utilisateurs sont très nombreuses et variées (Keith, 2005; Vanderdonckt, 1994) ;
- ne permet pas de couvrir tous les problèmes d'utilisabilité dans les systèmes interactifs (Farenc, 1997) ;
- problème de conformité entre les différentes règles si nombreuses dans la littérature (Mariage, 2005) ;
- les guides ergonomiques comprennent des règles qui sont exprimées en langage naturel et indépendamment de tout contexte d'utilisation (Keith, 2005; Rauterberg, 1996) ;

Intervenants : l'évaluateur.

Ressources matérielles : L'interface à évaluer.

Phase d'application dans le cycle de développement : peut être appliquée à partir de la fin de la phase de conception du système interactif : phase de test (ou encore sur un prototype de système interactif).

### Grille d'évaluation

Objectif : Permet de noter un système interactif suivant un ensemble d'aspects. Il s'agit d'un moyen efficace pour recueillir des données subjectives sur l'expérience de l'utilisateur avec un système interactive (Fallman et Waterworth, 2005).

15. L'évaluation des systèmes interactifs en utilisant les règles ergonomiques est développée d'avantage dans le chapitre 2.

Description : Afin de procéder à l'évaluation, l'évaluateur doit disposer d'une grille. Les lignes de la grille correspondent à des aspects que l'évaluateur note un par un (sur une échelle de 0 à 4 par exemple).

Avantages :
- Efficacité et simplicité de la technique ainsi que du processus d'évaluation proposée et,
- le faible coût du processus d'évaluation adopté.

Inconvénients :
- Il n'est pas évident que les aspects pris en considération couvrent la totalité de l'interface du système interactif et,
- relève d'un jugement purement objectif de la part de l'évaluateur.

Intervenants : l'évaluateur et l'utilisateur (facultatif).

Ressources matérielles : aucune ressource matérielle n'est requise pour le processus d'évaluation.

Phase d'application dans le cycle de développement : phase finale (phase de test).

#### 1.3.3.3 Techniques de test des systèmes interactifs

**Think Aloud Protocol (Jaaskelainen, 2000; Olmsted-Hawala *et al.*, 2010)**

Objectif : Connaître ce que pense l'utilisateur en interagissant avec le système.

Description : *Think Aloud Protocol* consiste à observer l'utilisateur interagissant avec le système interactif. Lors de l'interaction, l'utilisateur exprime ses pensées et ses réflexions. Il peut être assisté par une série de questions. Les pensées peuvent être enregistrées sur une vidéo ou encore en audio pour faciliter leur analyse et exploitation (Jaaskelainen, 2000).

Avantages :
- facile à mettre en place ;
- rapide ;
- se base à la fois sur l'observation et l'interview avec l'utilisateur lors de l'interaction et,
- non coûteuse.

Inconvénients :
- L'analyse des informations détectées est une tâche difficile à élaborer,
- Farenc indique que l'utilisateur peut être influencé par la manière de poser les questions (Farenc, 1997) et,

– le fait que l'utilisateur interagit avec le système et commente en même temps peut le perturber durant l'interaction ;

Intervenants : l'évaluateur et l'utilisateur.

Ressources matérielles : le système à évaluer.

Phase d'application dans le cycle de développement : phase de test.

**Oculométrie (Eye Tracking) (Beckers et Fuhr, 2010)**

Objectif : Permet de détecter, d'enregistrer et d'analyser le mouvement des yeux de l'utilisateur interagissant avec le système afin de l'exploiter pour l'évaluation. Cette technique permet de suivre le mouvement du regard, sa fixation ainsi que sa localisation (Beckers et Fuhr, 2010).

Description : Cette technique consiste à utiliser un dispositif spécial : l'oculomètre. Ce dispositif permet de détecter les mouvements des yeux de l'utilisateur ainsi que sa fixation du regard. Cette détection vise à déterminer les zones d'intérêt de l'*IHM* sur lesquelles l'utilisateur s'est focalisé (Beckers et Fuhr, 2010).

Avantages :

– Fiabilité des résultats issus de l'oculométrie pour l'évaluation ;
– les résultats issus de la capture du mouvement du regard de l'utilisateur sont généralement présentés sous forme visuelle ainsi nous relevons une facilité au niveau de leur exploitation pour l'évaluation et,
– avec les nouvelles avancées technologiques, les dispositifs matériels utilisés pour la capture de la direction du regard ne présentent aucune gène de l'utilisateur.

Inconvénients :

– l'analyse des résultats requiert un haut niveau d'expertise ;
– le coût élevé de l'oculométrie ;
– le dispositif de capture entraine une sorte de gêne coté utilisateur interagissant avec le système interactif via son interface utilisateur et,
– limitation du champ de visé du fait que le dispositif de capture requière parfois une fixation de la tête de l'utilisateur (Trabelsi, 2006; Chalon, 2004).

Intervenants : l'évaluateur, un expert en oculométrie et l'utilisateur.

Ressources matérielles : : le système à évaluer et un oculomètre.

Phase d'application dans le cycle de développement : phase de test.

**Mouchard électronique**

Objectif : Capturer les données portant sur l'*IHM* dans le but de les analyser afin de détecter les problèmes d'utilité (Grammenos *et al.*, 2000).

Description : Le mouchard électronique est défini par Ezzedine et al. comme étant un outil logiciel permettant le recueil en situation réelle des actions de l'utilisateur et leur effet sur le système lors de l'interaction avec le système évalué. L'évaluation des interfaces utilisateur s'effectue suivant deux phases. La première concerne la capture des données d'interaction entre l'utilisateur et le système interactif. La seconde phase est dédiée à la classification et l'analyse de ces données (Ezzedine *et al.*, 2006).

Avantages :
- La collecte d'information se fait d'une façon discrète et transparente du point de vue utilisateur. De ce fait, en aucun cas l'utilisateur ne se sent gêné par la présence du mouchard (Ezzedine *et al.*, 2006; Bastien et Scapin, 2002) ;
- le processus de recueil des données est généralement automatisé et,
- se base sur des données d'évaluation objectives, d'où une meilleure fiabilité des résultats d'évaluation.

Inconvénients :
- La quantité importante de données enregistrées pour l'évaluation (Bastien et Scapin, 2002) et,
- la complexité de la mise en place pour le processus d'évaluation ;

Intervenants : l'évaluateur, le calculateur et l'utilisateur.

Ressources matérielles : le système à évaluer.

Phase d'application dans le cycle de développement : phase de test ou encore avec un prototype avancé du système interactif.

#### 1.3.3.4 Techniques de prototypage

**Tests de conception**

Objectif : Vise à élaborer un système interactif avec un cycle itératif de conception (Trabelsi, 2006).

Description : D'après Senach, les tests de conception s'articulent essentiellement autour de trois étapes. La première consiste à sélectionner les différentes alternatives de conception du système interactif. Ensuite, des prototypes sont réalisés afin qu'ils soient testés lors de la troisième phase (Banc d'essai final) pour déterminer la meilleure alternative de conception du système interactif (Senach, 1990).

Avantages :
- intègre l'utilisateur lors du processus d'évaluation et,
- le cycle itératif d'évaluation que les tests de conception proposent, permet de détecter le maximum de problèmes liés à l'utilité et l'utilisabilité.

Inconvénients :
- Cette technique s'avère assez gourmande en matière de temps et de ressources ;
- au niveau du choix des alternatives de conception, les évaluateurs peuvent être confrontés à un grand nombre d'alternatives ;
- il n'existe pas de tests standards au niveau du banc d'essai final et,
- cette technique reste théorique et peu utilisée (Trabelsi, 2006).

Intervenants : l'évaluateur, le concepteur et l'utilisateur.

Ressources matérielles : spécification du système, maquette du système, environnement d'utilisation du système à évaluer.

Phase d'application dans le cycle de développement : Depuis les premières phases de conception.

### 1.3.4 Discussion

Lors de cette section nous avons introduit brièvement une liste de techniques d'évaluation. L'évaluation des *IHM* est un domaine riche en concepts et en techniques. De ce fait, avant de procéder à l'évaluation, l'évaluateur trouve des difficultés à choisir la technique appropriée. Chacune des techniques proposées a ses spécificités et traite un ensemble donné d'aspects à prendre en considération lors de l'évaluation. Certains auteurs tels que *Nielsen et Molich* recommandent le recours à plus d'une technique d'évaluation dans le but d'améliorer la qualité du résultat (Nielsen, 1993; Molich *et al.*, 1999). Dans ce cas, l'évaluateur sera confronté à un autre problème qui concerne la divergence des résultats d'évaluation. En effet, il n'est pas évident que les résultats issus de différentes techniques d'évaluation convergent. Par exemple ; l'interface d'un système interactif peut être conforme à des recommandations ergonomiques mais elle présente de nombreux problèmes lors de son utilisation. Alors que l'inspection avec les guides ergonomiques indiquent l'absence de problème de conception, une évaluation à l'aide du mouchard indique qu'il y a une multitude de problèmes d'utilisation. L'évaluateur peut ainsi remédier aux problèmes détectés par le mouchard afin d'améliorer le système interactif évalué. Une seconde évaluation peut inverser les résultats via l'indication de problèmes avec les guides ergonomiques et l'absence de problèmes avec le mouchard électronique. De ce fait, le couplage entre les différents résultats s'avère une tâche difficile à élaborer.

De plus, les techniques proposées différent suivant le niveau de l'automatisation du processus d'évaluation. Certaines peuvent être totalement automatisées, d'autres requièrent un traitement manuel assez important. Vu la complexité des nouvelles interfaces utilisateur et le nombre importants d'informations et de composants graphiques qu'elles contiennent, un grand nombre d'informations est manipulé pour l'évaluation de l'*IHM*.

De ce fait, un besoin d'automatiser le processus d'évaluation est nécessaire. L'automatisation permet de gagner un temps considérable par rapport à l'évaluation manuelle ou semi-automatique. Ainsi, elle augmente la fréquence possible des évaluations. D'ailleurs, elle permet plus de test unitaire et par conséquence plus de problèmes découverts dans l'interface évaluée (Farenc, 1997). Par ailleurs, l'automatisation de l'évaluation permet une « *démocratisation* » de l'évaluation. En effet, elle permet une accessibilité de l'évaluation par les non-experts du domaine et par la suite l'évaluation devient plus récurrente dans les pratiques des concepteurs de systèmes interactifs.

Un autre aspect est que la majorité des techniques d'évaluation s'appliquent lors de la dernière phase du cycle de conception des systèmes interactifs. Ceci peut être interprété d'une part par le fait que certaines techniques d'évaluation (telles que le mouchard électronique ou encore l'observation) se basent sur l'expérience utilisateur d'où la nécessité de disposer d'un système opérationnel afin de procéder à l'évaluation. D'autres techniques telles que l'inspection de la conformité de l'interface utilisateur par rapport à des règles ergonomiques requièrent des interfaces déjà réalisées ou encore des interfaces en cours de prototypage. L'évaluation, ainsi, n'est pas prise en considération depuis les premières étapes de conception des systèmes interactifs. Comme elle est reportée à la fin du cycle ; généralement elle est négligée faute de temps et de ressources. Certains auteurs ont proposé des cycles de conception orienté évaluation tel est le cas du modèle *Nabla* (Kolski, 2001), modèle étoile (Hartson et Hix, 1989) ou encore le modèle en U (Abed, 2001).
Cette diversité de techniques d'évaluation constitue l'une des motivations majeures pour la présente thèse. En effet, nous avons l'intention de se baser sur plus d'une technique d'évaluation et d'automatiser au maximum le processus d'évaluation.

## 1.4 Difficultés de l'évaluation de l'*IHM*

Comme les interfaces Homme-Machine contribuent à rendre l'usage des systèmes interactifs plus facile et intuitifs, de nombreux travaux se sont adressés à ce sujet. Ceci est accentué par le fait que l'*IHM* est la partie apparente aux utilisateurs. Elle dissimule les dizaines de milliers de lignes du code source. L'évaluation des *IHM* relèvent certaines difficultés que la communauté de l'*IHM* essaie de contourner par la mise en place d'outils, techniques et approches d'évaluation. Ces difficultés sont :

– face à la panoplie des nombreux outils et techniques d'évaluation, l'évaluateur se trouve souvent devant plusieurs choix pour procéder à l'évaluation. Chaque technique possède sa spécificité et traite un ensemble donné d'aspects (cf. Section 1.3) (Bark et Gulliksen, 2006) ;

- tout comme dans le génie logiciel, il n'y a pas de démarche canonique pour l'évaluation des *IHM* (Pressman, 2001) ;
- absence de méthodologie et de Framework pour l'évaluation des *IHM* (Seffah et Habieb-Mammar, 2009) ;
- l'évaluation des *IHM* requiert souvent une modélisation de cette dernière avec un haut niveau d'abstraction afin de bien gérer et exploiter les événements du système et de l'interface ;
- les données relatives à l'évaluation des *IU* suivant le critère d'utilisabilité sont très variés et hétérogènes (Schmettow, 2008)
- durant la phase de l'évaluation, l'évaluateur est souvent amené à faire face à de nombreuses informations hétérogènes. L'hétérogénéité de ces informations rend le processus d'évaluation assez délicat et périlleux à mettre en place ;
- il existe différents critères d'évaluation. En se basant sur tous ces critères, le processus d'évaluation devient une tâche trop difficile et coûteuse (en matière de temps et de ressources). De ce fait, il est préférable d'opter pour un ensemble donné de critères. D'où la difficulté au niveau du choix des critères pris en considération pour l'évaluation. A noter que chaque critère traite un ensemble donné d'aspects dans le système à évaluer ;
- l'absence de formalismes et de standards modélisant le processus d'évaluation (Seffah et Habieb-Mammar, 2009) ;
- les formalistes et les modélisations des résultats d'évaluation restent encore peu élaborés et travaillés ;
- l'évaluation, dans la plus part des cas, repose sur des jugements d'ordre subjectif de la part de l'évaluateur. En effet, ce dernier est souvent amené à interpréter les résultats et les données de l'évaluation. De ce fait, les résultats d'évaluation d'un système d'évaluation varient d'un évaluateur à un autre en procédant avec la même technique d'évaluation (Nielsen, 1993; Jeffries *et al.*, 1991; Molich *et al.*, 1999) et,
- Comme l'*IHM* est un domaine adjacent à un ensemble de disciplines (tel est le cas du génie logiciel, l'ingénierie, l'intelligence artificielle, linguistique, psychologie, ergonomie, sociologie, infographie, etc.), l'évaluateur doit souvent disposer d'un nombre important de connaissances de savoir-faire pour procéder à l'évaluation des interfaces Homme-machine (Vanderdonckt, 1999).

## 1.5 Synthèse et conclusion

Nous avons vu dans ce chapitre un aperçu sur l'évaluation des *IHM* : concept, techniques d'évaluation et difficultés associées. L'évaluation des *IHM* est un domaine très riche en matière de travaux et de notions. Cet intérêt est justifié par le fait que l'évaluation des

*IHM* est une phase primordiale et indispensable pour l'obtention d'Interfaces Homme-Machine de qualité. Les techniques dans ce contexte sont variées et multiples. Bien que les techniques d'évaluation visent à identifier les problèmes dans les *IHM*, elles diffèrent suivant les dimensions qu'elles traitent (utilité, utilisabilité et accessibilité) pour l'évaluation des *IHM*. De plus, la nature des données pour l'évaluation varie d'une technique à une autre. Les techniques peuvent se baser sur des données subjectives ou encore objectives. De nombreux travaux visent à automatiser l'utilisation de ces techniques (Balbo, 1996; Stone *et al.*, 2005; Xiong *et al.*, 2006; Fiora *et al.*, 2008). Cette automatisation consiste à implémenter ces techniques sous forme d'outils pour l'évaluation automatiques des *IU*. Ceci à pour but de :

- augmenter le spectre des aspects évalués dans le système ;
- augmenter la cohérence et l'acceptabilité des résultats d'évaluation ;
- réduire la nécessité de recours à des spécialistes en évaluation et par la suite réduire le coût de l'évaluation ;
- réduire le temps requis pour l'évaluation et,
- collecter un nombre plus important de données pour l'évaluation et par conséquent augmenter la crédibilité du processus d'évaluation.

Les experts en évaluation recommandent l'utilisation de plusieurs techniques pour l'évaluation des *IHM* (Nielsen, 1993; Molich *et al.*, 1999). En effet, les techniques peuvent se compléter pour couvrir le maximum de types de problèmes pouvant être détectés dans une *IHM*. Les techniques d'évaluation sont généralement implémentées sous forme d'outils.

Dans le prochain chapitre, nous allons aborder une revue de littérature des outils existants pour l'évaluation des systèmes interactifs en se basant sur les connaissances ergonomiques. Ces outils sont liés à la catégorie de l'inspection de la qualité ergonomique des interfaces utilisateur.

Chapitre 2

# Etat de l'art des outils d'évaluation des systèmes interactifs exploitant les connaissances ergonomiques

*« L'ergonomie des produits informatiques ? Au-delà de la fiabilité et de la performance, les utilisateurs réclament toujours plus de confort d'utilisation et d'efficacité de l'interface...*
*à juste titre ! »*

Gilles Murawiec

## Introduction

L'ÉVALUATION de la partie interactive (*IHM*) des systèmes interactifs constitue une étape importante du processus de conception de ces systèmes. Cette évaluation peut être considérée comme étant la validation de la conformité des interfaces par rapport à un ensemble de connaissances liées à l'ergonomie. Bien que les travaux liés à l'évaluation des *SI*soient abondants, nous relevons tout de même beaucoup de problèmes relatifs à l'utilité, l'utilisabilité et l'accessibilité des *SI*. Par exemple dans le cas du web, 90% des sites web comportent des problèmes d'utilisabilité (Research, 1999) et 66% d'entre eux sont inaccessibles pour les utilisateurs ayant un handicap (Jackson-Sanborn *et al.*, 2002). Lors du premier chapitre de ce mémoire, nous avons introduit les différentes techniques d'évaluation. Dans ce chapitre, nous nous focalisons sur les outils exploitant les règles ergonomiques pour l'évaluation et/ou l'aide à l'évaluation des SI. Le présent chapitre fait

l'objet d'un état de l'art de ces outils. Tout d'abord, nous nous intéressons à l'évaluation des *SI* en se basant sur les connaissances ergonomiques. L'objectif de cette première section est d'introduire les notions relatives à l'ergonomie logicielle. La deuxième section présentera une classification dédiée aux outils d'évaluation des SI. Cette section servira pour la présentation des différents outils existants lors de la troisième section. Afin de compléter ce chapitre, la quatrième section comportera une synthèse. L'objectif de ce chapitre et de mettre en évidence les outils existants pour l'évaluation des systèmes interactifs (en se basant sur des connaissances ergonomiques), et ce, dans le but de positionner ces travaux par rapport au notre présenté dans le cinquième chapitre.

## 2.1 Des notions sur l'ergonomie

### 2.1.1 L'ergonomie

Il est connu depuis longtemps que l'un des principaux facteurs de qualité des systèmes interactifs est l'interface du système elle-même (Baggen *et al.*, 2011). Il existe une grande variété d'outils et des méthodes pour aider le concepteur à réaliser des *IHM* utiles et utilisables (Alter, 2010; Rogers *et al.*, 2011). Chaque méthode et chaque outil a ses spécificités, ses propres exigences et se base sur une démarche qui lui est propre. De plus, ces outils et méthodes peuvent couvrir différents problèmes (Morandini *et al.*, 2011). Certains outils consistent à faire des tests utilisateurs dans le but d'identifier le maximum de problèmes d'utilité et d'utilisabilité de l'interface à évaluer. Néanmoins, ces méthodes et outils requièrent une grande expertise de la part des évaluateurs. De plus, ils sont aussi coûteux (en matière de temps et de ressources matérielles) (Scapin *et al.*, 2002). D'autres outils s'appuient sur la validation de la conformité de l'*IHM* par rapport à des règles ergonomiques formalisées (issues de travaux en ergonomie logicielle). L'ergonomie est définie comme étant la mise en oeuvre de l'ensemble des connaissances scientifiques relatives à l'homme et nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être utilisés par le plus grand nombre d'utilisateurs avec le maximum de confort, de sécurité et d'efficacité (Seffah et Metzker, 2008). L'ergonomie est un domaine pluridisciplinaire adjacent à beaucoup de domaines : ergonomie logicielle, physiologie, psychologie cognitive, etc. (Vanderdonckt, 1994).

L'ergonomie vise à améliorer la communication entre opérateur humain et système interactif. Pour ce qui est du domaine qui nous intéresse dans ce chapitre, les travaux en ergonomie logicielle sont nombreux et datent de plus d'une quarantaine d'années. Généralement, les connaissances ergonomiques sont exprimées sous forme de règles ergonomiques, utilisables pour la conception et l'évaluation de logiciels interactifs.

### 2.1.2 Règle ergonomique

Une règle ergonomique constitue, selon Vanderdonckt, un principe de conception et/ou d'évaluation à observer en vue d'obtenir et/ou de valider une interface Homme-Machine ergonomique (Vanderdonckt, 1999). Elle peut être issue de travaux provenant d'autres disciplines, telles que le génie logiciel, ou se base sur l'observation ou l'étude du comportement d'utilisateur des systèmes interactifs. Les règles ergonomiques sont exprimées en langage naturel afin de guider le concepteur et/ou l'évaluateur pour l'obtention d'interfaces utiles, accessibles et facilement utilisables.

Nous prenons comme exemple de règles ergonomiques, celle extraite du manuel de *Philippe Kortum* : « *HCI Beyond the GUI* » (Kortum, 2008, page 249) énonçant qu'un message d'erreur doit être composé de trois parties :

1. Une indication sur l'erreur rencontrée
2. Un aperçu de l'erreur rencontrée
3. La façon avec laquelle l'utilisateur doit réagir afin de remédier au problème rencontré (dans le cas où le problème est rectifiable) et les éventuelles possibilités que l'utilisateur peut opter.

Dans la figure ci-dessous, un exemple de deux messages d'erreurs, Fig 2.1. Le premier concerne un message où il est difficile (voire même impossible) de déterminer la source de l'erreur, Fig.2.1.a. Le second exemple, quant à lui, indique un message d'erreur contenant clairement la source du problème (nom manquant), le résultat (échec de l'enregistrement) et les solutions proposées (entrer le nom et de re-procéder à l'enregistrement)Fig.2.1.b.




2.1.a. Exemple négative




2.1.b. Exemple Positif

FIGURE 2.1: Exemples positif et négatif de messages d'erreur.

### 2.1.3 Critère ergonomique

Dans la littérature, il existe une grande variété de règles ergonomiques (par exemple 3700 règles ergonomiques sont décrites dans (Vanderdonckt, 1994)). Afin de faciliter l'usage et la consultation des règles ergonomiques, plusieurs recherches ont porté sur leur classification dans des catégories intitulées critères ergonomiques. A titre d'exemple,

le tableau 2.1 illustre les dix critères proposés par *Mauney* et *Masterton* (Mauney et Masterton, 2008) portant sur les interfaces à petits écrans [1].

| **Critère** | **Métrique** |
|---|---|
| **Apprentissage** | Une interface doit être facile à être utilisée dès la première utilisation. |
| **Efficacité** | Le nombre d'étapes pour accomplir une tâche donnée par un utilisateur. |
| **Mémorisation** | Une interface peut ainsi être facile à manipuler chaque fois où l'utilisateur interagit avec elle. |
| **Récupération d'erreur** | Dans une interface utilisateur, les utilisateurs ne devraient jamais être autorisés à commettre une erreur. |
| **Simplicité** | Les tâches courantes doivent être faciles à exécuter et les tâches les moins courantes doivent être possibles. |
| **Mapping** | Ce que l'utilisateur s'attend, quand il interagit avec l'interface, doit avoir lieu. |
| **Visibilité** | Les informations importantes doivent être les plus visibles sur l'interface (et inversement les informations les moins importantes). |
| **Retour d'information** | L'utilisateur doit toujours garder le contrôle sur l'interface. |
| **Satisfaction** | Le niveau de l'appréciation de l'utilisateur de l'interface. |
| **Cohérence** | Les items similaires doivent être affichés de la même façon sur les mêmes IHM. |

TABLE 2.1: Les dix critères ergonomiques proposés par Mauney et Masterton pour les interfaces à petits écrans (Mauney et Masterton, 2008)

### 2.1.4 Evaluation de la qualité ergonomique des systèmes interactifs

Nous entendons par évaluation de la qualité ergonomique des systèmes interactifs, la vérification de la conformité du système à évaluer, relativement à un ensemble de règles ergonomiques. L'évaluation ergonomique vise à détecter les aspects de l'interface qui peuvent engendrer des difficultés d'utilisation ou bien des réductions de performances du

1. Les interfaces à petits écrans (Small-Screen Interfaces) : il s'agit généralement des périphériques PDA ou bien terminaux de téléphones portables. Généralement, la question qui se pose pour la conception de systèmes interactifs opérant sur de telles interfaces est comment afficher les informations d'une façon visible et bien claire (Mauney et Masterton, 2008)

système Homme-Machine. Ainsi, l'évaluation constitue un apport essentiel pour la mise en place de systèmes interactifs utiles et facilement utilisables par différents utilisateurs. Donc, elle s'avère fondamentale pour l'amélioration de la qualité des systèmes. L'usage des critères et des règles ergonomiques a montré son intérêt lors de l'évaluation des systèmes interactifs (Vanderdonckt, 1994). Néanmoins leur application révèle beaucoup de difficultés :

- Les connaissances ergonomiques sont très souvent énoncées en langage naturel. De ce fait, les règles ergonomiques sont souvent difficiles à interpréter par l'évaluateur (Keith, 2005) ;
- les règles ergonomiques sont souvent générales et indépendantes de tout contexte d'usage (Grammenos *et al.*, 2000; Rauterberg, 1996) d'où une interprétation contextuelle concrète s'impose (Vanderdonckt, 1995). Ceci rend l'exploitation des règles ergonomiques pour l'évaluation assez difficile (Beirekdar, 2004). Les résultats trouvés peuvent varier quand différents évaluateurs étudient la même *IHM* même avec les mêmes techniques (Ivory et Hearst, 2001; Scapin *et al.*, 2002; Charfi *et al.*, 2009) ;
- certains critères ne sont pas quantifiables en termes d'objectif mais plutôt relèvent des jugements quantitatifs d'où la difficulté de les inspecter dans l'interface évaluée. En effet, certaines règles traitent par exemple de la qualité des messages d'erreurs et des instructions pour l'utilisateur. L'évaluation de telles règles ne peut pas avoir lieu d'une façon automatisée. L'intervention d'un expert humain s'avère indispensable dans le processus d'évaluation et requiert des jugements subjectifs ;
- les règles ergonomiques sont difficiles à gérer vu leur nombre et l'absence de conformité dans leur présentation (Grammenos *et al.*, 2000; Trabelsi, 2006; Mariage, 2005). Il est bien difficile d'identifier une règle dans une « *jungle* » de règles (Mariage et Vanderdonckt, 2005). Elles sont souvent réparties dans plusieurs sources de diverses natures (Vanderdonckt, 1999). Nous citons essentiellement :
  - les ensembles de règles ergonomiques : ils consistent en une collection de règles proposées pour l'obtention d'*IHM* de qualité. Exemple : « *le Guide Ergonomique des interfaces homme-machine* » (Vanderdonckt, 1994), règles ergonomiques pour le web de (Rukshan et Baravalle, 2011), « *les critères ergonomiques pour les interactions homme-environnements virtuels* » proposés dans (Bach et Scapin, 2005). Il existe certains guides ergonomiques spécifiques à un domaine d'application donné tel que « *Electoral ergonomic guidelines* » (Michel *et al.*, 2007) ou encore des recommandations ergonomiques pour la conception de systèmes d'assistance cognitive dans les habitats (Serna *et al.*, 2010) ;

- les standards de conception : ensemble de spécification pour l'obtention d'*IHM* standards ; exemple : la norme *ISO 9241* (ISO, 1998) et le *WCAG*[2](Kapsi *et al.*, 2009) ;
- les guides de style : le guide de style est un outil bien répandu pour définir la présentation d'une *IHM* au sein d'une même plateforme ou d'une même ligne de produits (Mariage, 2005). A titre d'exemple, citons le guide ergonomique proposé par Java « *Java Look and Feel Design Guidelines* » (Sun Microsystems et Javasoft, 1999), le guide ergonomique des applications sous Android (Android, 2012) ou encore le guide proposé par Microsoft intitulé « *Windows User Experience Interaction Guidelines* » (Microsoft, 2011) et,
- les algorithmes de conception ergonomiques : ils proposent des démarches et des méthodes à suivre pour obtenir des *IHM* de bonne qualité ergonomique. Exemple : « *Design Adviser* » (Faraday, 2000).

– l'application des règles ergonomiques reste une tâche complexe qui requiert des bonnes connaissances, voire une expertise élevée (Farenc *et al.*, 2001). Bien que les connaissances semblent être faciles à passer en revue, nous relevons une difficulté qui réside essentiellement dans le fait que les règles ergonomiques requièrent des connaissances variées dans plusieurs domaines ;
– l'ergonomie est un domaine adjacent à plusieurs domaines (cognitifs, génie logiciel, psychologie, etc.) (Vanderdonckt, 1999). C'est pour cela que les termes utilisés pour la description des règles ergonomiques, sont difficiles à comprendre et à interpréter par de nombreux concepteurs et programmeurs (Scapin *et al.*, 2002) et,
– comme en génie logiciel, nous relevons dans l'ergonomie logicielle l'absence de démarche canonique pour la conception et le développement d'*IHM* d'où l'absence de solution ergonomique universelle (Sperandio, 1996). L'ergonomie logicielle n'est pas le fait de trouver une solution ou bien un algorithme dans le but d'accomplir une tâche donnée.

Les outils d'évaluation des *IHM* (point de vue ergonomie logicielle) se différencient principalement par les critères ergonomiques qu'ils inspectent dans les interfaces comme précisé précédemment. La littérature en matière de connaissances ergonomiques est fort abondante. En effet, nous retrouvons de nombreux documents aux contenus très hétérogènes. Nous pouvons citer à titre d'exemples : les 247 règles ergonomiques de Travis portant sur le web (« *Web Usability Guidelines* ») (Travis, 2011) ou encore les 113 règles proposées par Nielsen « *Top ten Guidelines for Homepage Usability* » (Nielsen et Tahir, 2001). Ces outils se différencient aussi par l'instant de leur application lors du cycle de conception d'une *IHM*. En effet, certains outils consistent à assister le concepteur lors de

2. Web Content Accessibility Guidelines : Ensemble de recommandations issues du W3C destiné au programmeur web afin de garantir dans leurs sites une meilleure accessibilité surtout pour les utilisateurs handicapés et les seniors. Le *WCAG* comporte essentiellement 14 directives (WCAG, 2011).

la spécification et la conception du système en prenant en compte des critères ergonomiques pour développer une interface utilisateur de bonne qualité ergonomique. Nielsen estime qu'il revient 100 fois plus cher d'établir des corrections du système déjà réalisé que de le corriger au début du cycle de développement (Nielsen, 1993). Nous citons à titre d'exemple les tests heuristiques de l'interface en cours de conception. D'autres outils intègrent les connaissances ergonomiques pour l'évaluation des systèmes interactifs dans leur état final ou encore sous forme de maquette ou encore de prototypes plus ou moins avancés du système. Dans ce chapitre nous nous intéressons à la catégorie relative à l'inspection de la conformité de l'interface par rapport à un ensemble de règles ergonomiques. Certains de ces outils fournissent une évaluation automatique de l'interface. D'autres se contentent d'assister l'évaluateur lors de la phase d'évaluation. Ces travaux sont nombreux. De ce fait, une classification de ces outils est proposée dans la section qui suit.

## 2.2 Classification des outils d'évaluation des systèmes interactifs exploitant les connaissances ergonomiques

Plusieurs classifications des outils d'évaluation des *IHM* existent. Certaines se basent sur le degré d'automatisation de la méthode comme étant un critère de classification (Ivory et Hearst, 2001). Senach se réfère à la nature des données pour l'évaluation : outils se basant sur des méthodes empiriques ou bien sur des méthodes analytiques (Senach, 1990). La classification de Bastien et Scapin distingue les méthodes en deux grandes catégories ; la première regroupe les méthodes qui requièrent la participation de l'utilisateur lors du processus d'évaluation ; la seconde quant à elle regroupe les méthodes qui ne nécessitent pas le recours à l'utilisateur lors de l'inspection de l'interface à évaluer (Bastien et Scapin, 2001). Il existe d'autres classifications : (Whitefield *et al.*, 1991; Grislin et Kolski, 1996), etc. (cf. Chapitre 1). Les classifications des méthodes d'évaluation sont nombreuses, mais il n'en existe aucune spécifique aux outils qui assurent l'évaluation de la qualité ergonomique des systèmes interactifs. En effet, vu l'importance de l'ergonomie logicielle pour la conception et l'évaluation des systèmes interactifs, beaucoup d'outils ont été mis en place. De ce fait, nous estimons utile de classifier ces outils. Ces outils d'évaluation de la qualité ergonomique possèdent leurs spécificités. Ces spécificités s'articulent essentiellement autour des aspects qu'ils inspectent et les règles ergonomiques sur lesquelles ils se basent pour le processus d'évaluation. Nous proposons de classifier les outils d'évaluation suivant leur flexibilité pour l'ajout, la modification et la suppression de critères ou de règles ergonomiques pour le processus d'évaluation sans modifier la source du système d'évaluation. En effet, il nous parait utile de distinguer les différentes catégories ayant la spécificité de pouvoir alimenter la base par de nouvelles règles. En

effet, vu le nombre important de règles ergonomiques, il est difficile voire même impossible d'opter pour un outil d'évaluation qui puisse prendre en considération toutes les règles. Par conséquent, il serait préférable de mettre au point un système d'évaluation configurable qui permet l'ajout, la modification et la suppression de règles ergonomiques sans autant toucher à la source du système d'évaluation ; autrement dit, nous nous intéressons plus aux outils dont les règles ergonomiques ne sont pas intégrées dans le moteur d'évaluation. Dans la plupart des outils d'évaluation, nous retrouvons les règles ergonomiques codées en dur dans le moteur d'évaluation : c'est ce qu'on appelle le codage en dur (*le Hard-Coding*). Dans notre classification, nous retrouvons principalement deux grandes catégories comme illustrées dans Figure 2.2.

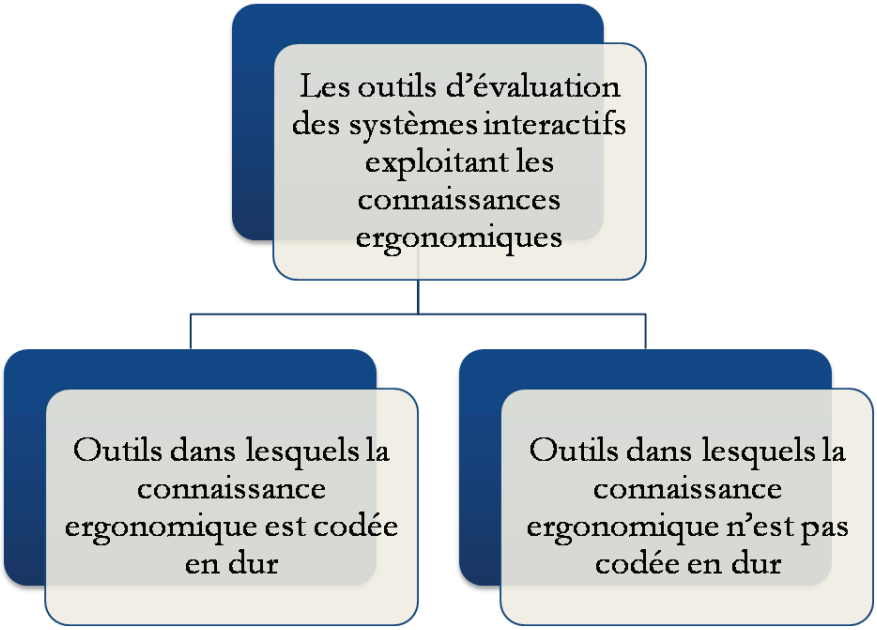



FIGURE 2.2: Classification proposée des outils d'évaluation exploitant les connaissances ergonomiques.

Notons qu'il existe une catégorie d'outils baptisée *Tools For Working With Guidelines* (*TFWWG*). Les outils issus de cette catégorie sont conçus dans le but de formaliser les différentes règles standards et ergonomiques issues des différents guides de style. De plus, ces outils permettent de supporter les recommandations ergonomiques ainsi que les algorithmes ergonomiques. La motivation majeure de la mise en place de tels outils réside dans le fait que les concepteurs des systèmes interactifs disposent généralement dans les meilleurs des cas de connaissances limitées concernant les règles ergonomiques. Ainsi, ce genre d'outils sont dédiés à inciter les concepteurs à prendre en considération les règles ergonomiques lors de la conception des différents systèmes interactifs. Comme exemple représentatif de ces outils, nous pouvons citer le projet Sierra qui comprend 3700 règles ergonomiques (Vanderdonckt, 1999). Les lecteurs intéressés peuvent consulter (Vanderdonckt et Farenc, 2001).

### 2.2.1 Les outils dans lesquels les connaissances ergonomiques sont codées en dur

La majorité des outils existant pour l'évaluation des systèmes interactifs se restreignent à un ensemble donné de critères ergonomiques pour l'évaluation des systèmes interactifs. L'évaluateur dans ce cas se trouve limité par des critères et un contexte d'évaluation imposés par l'outil. Ces outils sont généralement spécifiques à des systèmes interactifs particuliers. En effet, les règles ergonomiques sont souvent générales et indépendantes de tout contexte d'usage (Rauterberg, 1996). Lors de leur usage, les évaluateurs procèdent à une interprétation contextuelle d'une façon concrète en fonction du système interactif qu'ils évaluent (Keith, 2005; Vanderdonckt, 1995). Ces interprétations diffèrent d'un système interactif à un autre. Par conséquent les résultats trouvés à l'aide de ces outils sont peu fiables pour l'évaluation des systèmes interactifs. Afin de procéder à l'évaluation, l'évaluateur doit généralement sélectionner un ensemble de règles requises pour l'évaluation suivant le contexte d'évaluation et le système interactif à évaluer.

### 2.2.2 Outils dans lesquels les connaissances ergonomiques ne sont pas codées en dur

Cet ensemble d'outils offre à l'évaluateur la possibilité de sélectionner et de paramétrer le processus d'évaluation des systèmes interactifs sans modifier le code source de l'outil. Il est donc plus utile de spécifier l'ensemble des critères et règles ergonomiques qui vont être inspectés. Cela évite l'inspection de certaines règles qui ne sont pas utiles pour l'évaluation d'un système interactif quelconque. Ceci permet en conséquence un gain important en temps et en ressources en évitant d'évaluer l'interface Homme-Machine par rapport à certains critères dans le contexte concerné. En effet, certains systèmes interactifs s'adressent à une population d'utilisateurs bien spécifique. L'application de certains critères implémentés dans l'outil peut s'avérer inutile pour de tels systèmes ; ils peuvent faire appel à des critères spécifiques pour cette population. A titre d'exemple, l'outil Destine (Mariage *et al.*, 2004; Beirekdar *et al.*, 2002) permet de procéder à l'évaluation grâce à un langage de définition de règles ergonomiques, d'intégrer ou bien modifier des règles destinées au processus d'évaluation sans altérer le code source de l'outil d'évaluation. Nous pouvons classifier ces outils en deux sous-catégories, Figure 2.2. La première sous-catégorie comprend l'ensemble des outils qui permettent la sélection des règles ergonomiques à prendre en considération lors du processus d'évaluation. Ceci permet d'éviter d'inspecter la conformité de certaines règles inutiles pour certains systèmes interactifs. De plus l'évaluateur peut modifier le contenu de certaines règles lors de l'évaluation.

Néanmoins, cette catégorie d'outils ne permet pas l'ajout de nouvelles règles ergonomiques pour l'évaluation sans modifier le code source de l'outil. La seconde sous-catégorie concerne l'ensemble des outils permettant l'ajout, la modification et la suppression de critères ergonomiques sans altérer le code source du système d'évaluation. Les règles ergonomiques ainsi que les algorithmes d'inspection ne sont pas codés en dur dans le moteur d'évaluation. Dans ce qui suit, nous exposons les outils logiciels pour l'évaluation ergonomique des systèmes interactifs exploitant les connaissances ergonomiques.

## 2.3 Etat de l'art des outils d'évaluation des *IHM* exploitant les connaissances ergonomiques

Plusieurs outils ont été conçus en vue d'évaluer les systèmes interactifs. Dans cette partie, nous allons présenter ces outils. Plusieurs aspects seront présentés pour chaque outil. La présentation de chaque outil traitera les points suivants :

- **Nature de l'outil.** Outil logiciel, application, web service, plugin à installer sous un environnement intégré de développement quelconque et application client-serveur.
- **Interface homme-machine visée.** Nous nous intéressons aux interfaces que l'outil peut évaluer : interface WIMP [3], interface web (WUI) (Nerurkar, 2001) ou interface Mobile (Monk *et al.*, 2004).
- **Le facteur de qualité évalué.** Les dimensions de qualité logicielle inspectées : l'accessibilité, l'utilité ou l'utilisabilité.
- **Type d'évaluation.** Le type des données sur lesquels va porter l'évaluation : statique (des attributs des composants graphiques de l'*IHM*), dynamique (portant sur l'interaction entre l'utilisateur et l'*IHM*) ou bien à l'appréciation de l'utilisateur.
- **Principe d'évaluation.** Le principe de l'évaluation adopté : confrontation entre un modèle de référence et un modèle d'objet (Senach, 1990), le White Box testing ou le Black Box testing.
- **Données de l'évaluation.** Les données utilisées pour l'évaluation : les interactions entre l'utilisateur et l'interface, le fichier log, l'event logging (Woo et Mori, 2004), code source ou tout simplement un descriptif de l'interface à évaluer.
- **Acquisition des données.** La technique avec laquelle l'outil acquiert les données évoquées précédemment pour l'évaluation est soulignée : parcours de code source, mouchard électronique, fichier de configuration, questionnaire (McNamara et Kirakowski, 2011), analyse de fichier log, description textuelle, robot ou parseur de code.

3. On entend par WIMP les interfaces logicielles des systèmes interactifs courants.

- **Service fourni par l'outil en fin de processus d'évaluation.** Le service fourni indique le résultat final de l'évaluation : compte rendu sur la qualité ergonomique, rapport sur les règles ergonomiques non respectées, correction automatique et/ou assistée des incohérences ergonomiques ou suggestion d'amélioration.
- **Les intervenants dans le processus d'évaluation.** Les intervenants dans le processus d'évaluation peuvent être : l'évaluateur, utilisateur(s) (expérimental(s) ou final(aux), expert d'utilisabilité, développeur, expert en ergonomie logicielle, concepteur.
- **Connaissance requise pour l'usage de l'outil.** Nous y exposons l'ensemble des connaissances requises pour l'usage de l'outil par l'évaluateur. Nous citons les connaissances en : tâches utilisateurs, ergonomie logicielle, système à évaluer, utilisateur, langage de programmation ou logiciels spécifiques.
- **Niveau d'automatisation.** En s'inspirant d'*Ivory* et *Hearst* (Ivory et Hearst, 2001), nous distinguons quatre niveaux d'automatisation :
  - Aucune : Le processus d'évaluation est totalement manuel. Dans ce processus, l'évaluateur collecte les informations d'évaluation et effectue les analyses pour déduire des rapports sur la qualité ergonomique et des critiques de l'interface.
  - Capture : L'outil d'évaluation capture les données nécessaires à l'évaluation sans intervention de l'évaluateur (temps d'exécution d'une tâche, les erreurs engendrées par l'utilisation, etc.).
  - Analyse : L'outil d'évaluation analyse les différentes données pour l'évaluation des *IHM*. L'analyse s'effectue par l'interprétation des données pour identifier les problèmes de l'utilisabilité dans l'interface évaluée.
  - Critique : Une fois que nous disposons des données enregistrées et analysées (que ce soit d'une façon automatique, semi-automatique ou encore manuelle), l'outil d'évaluation fournit à l'évaluateur un ensemble de critiques et de suggestions ainsi que des solutions pour corriger les problèmes d'*IHM* détectés dans le système interactif afin d'améliorer l'*IHM*.
- **Flexibilité de l'évaluation.** Permet de renseigner si l'outil permet à l'évaluateur de choisir les règles à inspecter, de configurer le rapport d'évaluation, ou encore de configurer le processus d'évaluation.

Les outils seront présentés en deux parties. La première comportera les outils ne codant pas en dur les connaissances ergonomiques (pas de codage en dur des *RE*). La seconde comporte l'ensemble d'outils dont les règles ergonomiques sont codées dans le moteur d'évaluation et ne permettent pas l'ajout, la modification et la suppression des règles ergonomiques sans toucher au programme source.

Les outils présentés sont organisés chronologiquement dans les deux catégories.

### 2.3.1 Les outils dont la connaissance ergonomique n'est pas codée en dur

Dans cette section, des outils d'évaluation des systèmes interactifs seront présentées. Ces outils ont l'avantage de permettre le choix, l'ajout, la modification et la suppression des règles ergonomiques dans le processus d'évaluation des systèmes interactifs. Ainsi, cette catégorie d'outils offre une grande flexibilité pour le processus d'évaluation. Son point fort est l'indépendance de l'outil d'évaluation par rapport les systèmes interactifs à évaluer. En effet, l'évaluateur peut ajouter de nouvelles règles ergonomiques et supprimer les règles existantes suite à un changement du contexte.

#### 2.3.1.1 SYNOP (Kolski et Millot, 1991)

*SYNOP* est un système expert développé avec le langage *LISP*. Il permet l'évaluation statique des interfaces graphiques. L'évaluation s'effectue grâce à des descriptions des pages écrans fournies par un logiciel graphique : *IMAGIN*. L'évaluation se fait suivant la base des connaissances du système expert (chaque base utile pouvant être sélectionnée grâce à l'activation de méta-connaissances). La base de connaissances est l'ensemble de recommandations ergonomiques. SYNOP permet de détecter les erreurs dans l'interface à évaluer. Il modifie automatiquement ces défaillances et propose, en cas d'impossibilité de correction automatique, des suggestions d'amélioration. Il a été validé sur des *IHM* de supervision de systèmes industriels.

#### 2.3.1.2 Ocawa : Outil de Contrôle et d'Analyse pour le Web Accessible (Ocawa, 2011)

*Ocawa* est un plugin (un add-on) qui peut être installé sous « *Mozilla Firefox*[4] ». Ce plugin exploite un service web pour l'évaluation de l'accessibilité des sites web. Nous retrouvons trois versions d'*Ocawa* :

- une version logicielle téléchargeable gratuitement depuis le site web de l'éditeur (Ocawa, 2011). Avec cette version, il n'est possible que de procéder à un nombre limité d'utilisations par jour ;
- une version en ligne payante, contrairement à la précédente ne présente aucune restriction au niveau du nombre d'utilisations et,
- une version logicielle commerciale : elle consiste à installer une solution client-serveur dans un ou plusieurs postes. Elle supporte davantage de fonctionnalités pour l'évaluation des interfaces du type WUI.

4. Il n'est supporté que par les anciennes versions du navigateur : la version 1.9 est la plus récente de celles qu'*Ocawa* peut intégrer.

Ocawa est un système expert utilisant une base de faits et une base de connaissances. Les connaissances ergonomiques ne sont pas codées en dur, Figure 2.3. Dans la version gratuite, il nous a été impossible de modifier la base de connaissances par l'ajout de nouvelles règles. D'après l'éditeur, l'ajout, la suppression et la modification des règles ne se fait que dans la version payante en exprimant les règles avec OKWML [5] pour enregistrer ces règles sous forme de fichiers XML. Les règles intégrées dans la version gratuite sont une collection de règles issues des travaux de *WCAG 1.0* (WCAG, 2011) et *d'ADAE/DGME*[6]. Le processus d'évaluation est totalement automatisé. Il acquiert les données à partir du code HTML de la page à évaluer. L'évaluateur peut sélectionner l'ensemble de règles qu'il veut inspecter dans l'interface. La version, que nous avons testée, fournit en fin d'évaluation un rapport sur la page à évaluer indiquant les différentes incohérences détectées. Nous retrouvons dans Ocawa des filtres pour les données dans le rapport d'évaluation pour sélectionner les aspects à afficher.

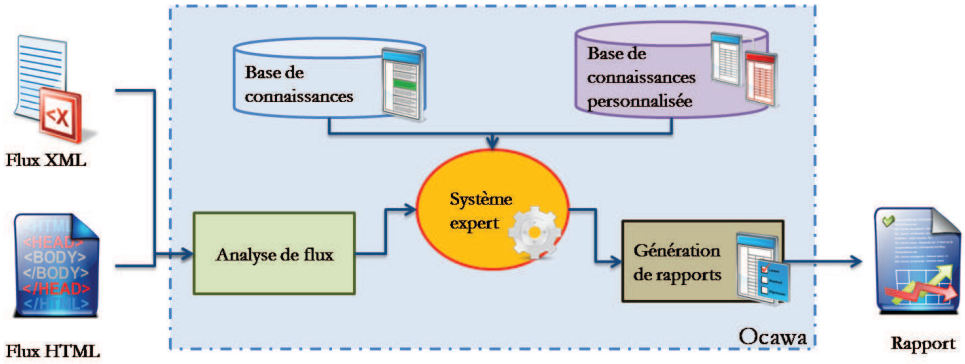



FIGURE 2.3: Architecture de l'outil Ocawa

#### 2.3.1.3 TAW : Test de Accesibilidad Web (TAW, 2011)

TAW permet de tester l'accessibilité d'un site web en se basant sur trois guides ergonomiques du web : *WCAG 1.0* (WCAG, 2011), *WCAG 2.0* (version beta) [7] et le *MobileOK*[8] (version beta). L'évaluateur peut alimenter les règles prises en considération pour l'évaluation du site via une interface graphique de saisie. Il peut aussi sélectionner les règles ergonomiques qu'il souhaite vérifier dans la page web à évaluer. TAW parcourt le code HTML de la page à évaluer. Le rapport d'évaluation consiste à marquer les différentes critiques par des icônes sur la page évaluée. Cet outil fournit à l'évaluateur les résultats

5. OKWML : un langage proposé par Ocawa pour l'écriture et la formalisation des règles ergonomiques (Ocawa, 2011).
6. ADAE/DGME : Agence pour le Développement de l'Administration Electronique.
7. Seconde version de WCAG parue en 2008 (WCAG, 2011).
8. Le W3C's Mobile OK Basic est une initiative du W3C dans le but d'aider les concepteurs à mettre en place des applications de bonne qualité du point de vue de l'*IHM* (W3C, 2008).

de l'évaluation sous forme de rapport détaillé en HTML et en EARL [9]. Ces deux rapports contiennent aussi l'ensemble de règles impossibles à vérifier automatiquement. De plus, TAW existe en plusieurs versions : logicielle (TawStandalone), service web (Taw online) etc. Le point fort de TAW est essentiellement son rapport qui est généré en EARL. Les résultats de l'évaluation sont ainsi facilement exploitables. Cependant TAW ne permet pas d'interpréter automatiquement des résultats de l'évaluation et ne fournit pas des suggestions pour l'amélioration de l'*IHM*. Il ne fournit que les aspects non conformes aux règles ergonomiques ainsi que les recommandations ergonomiques associées.

#### 2.3.1.4 Sherlock (Grammenos *et al.*, 2000)

Sherlock est une application client-serveur qui permet l'évaluation des interfaces WIMP en inspectant l'utilité et l'accessibilité des interfaces utilisateur. Il propose deux types d'évaluation : dynamique et statique. Dans la première, il inspecte l'interaction entre l'utilisateur et le système interactif. Quant à la seconde, il inspecte la présentation statique des informations sur l'interface. Le serveur contient les différentes règles ainsi que les algorithmes d'inspection de la qualité ergonomique. Les machines clientes envoient la description textuelle de l'interface à évaluer via le réseau au serveur. Ce dernier procède à l'inspection des règles ergonomiques dans cette description. Ensuite il avertit les machines clientes à travers un rapport d'évaluation, Figure 2.4. Sherlock évalue les facteurs ergonomiques et vérifie la cohérence entre les tâches et des propriétés visuelles. L'évaluation à l'aide de Sherlock implique clairement quatre intervenants : utilisateur, analyste d'utilisabilité, expert d'utilisabilité, concepteur d'interface (Grammenos *et al.*, 2000). Le processus d'évaluation ergonomique est automatisé.

9. EARL est un format lisible par la machine pour exprimer les résultats de tests. EARL est conçu pour faciliter le traitement des résultats des tests, tels que ceux générés par les outils d'évaluation d'accessibilité du web (W3C, 2011)

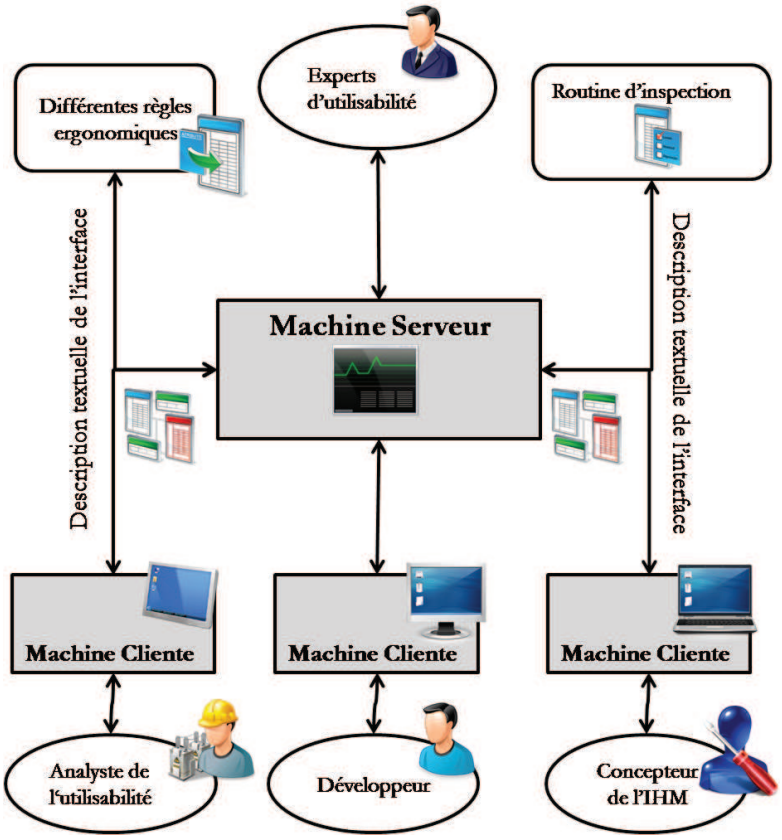



FIGURE 2.4: Architecture de Sherlock (Inspiré de (Grammenos *et al.*, 2000))

#### 2.3.1.5 ERGOVAL (Farenc *et al.*, 2001)

Ergoval est un système expert. Il se base tout d'abord sur la décomposition de l'interface évaluée suivant des objets (intitulés objets de présentation). Ces objets sont par la suite liés à des règles ergonomiques suivant des structures. Ensuite, la cohérence des objets est inspectée par rapport à l'ensemble de règles ergonomiques associées à travers le système expert. ERGOVAL fournit une aide pour l'inspection de la qualité ergonomique des *IHM*. Il évalue statiquement les interfaces WIMP. L'évaluation s'effectue en comparant les attributs des objets graphiques de l'interface avec un ensemble de recommandations ergonomiques. Le système se base dans cette comparaison sur deux stratégies d'évaluation. La première stratégie se base sur la structure (en profondeur), elle consiste à évaluer l'interface objet par objet. La seconde (transversale) évalue les interfaces utilisateur critère par critère. Afin de faciliter l'exploitation des règles ergonomiques pour l'évaluation de l'*IHM*, celles-ci sont reformulées. Elles sont classifiées suivant les formes des objets, le placement des objets, etc. Ensuite, l'évaluateur assure le lien entre les objets graphiques de l'interface avec les règles ergonomiques à vérifier. Le processus d'évaluation n'est automatisé qu'au niveau de la capture et de l'analyse. ERGOVAL offre une flexibilité au niveau de la configuration de règles ergonomiques et de liaison entre cette dernière

et les attributs graphiques. Cet outil ne fournit aucune interprétation ou suggestion à l'évaluateur. Il ne permet pas de traitement automatique des informations capturées.

#### 2.3.1.6 Destine : Design and Evaluation STudio for INtent based Ergonomic web sites (Beirekdar *et al.*, 2002; Beirekdar, 2004)

Destine est un environnement pour l'évaluation statique des sites web en inspectant leur utilité et leur accessibilité. Le processus d'évaluation se déroule comme suit : tout d'abord l'évaluateur doit structurer les règles ergonomiques. Ensuite, il planifie l'évaluation. A la fin du processus d'évaluation, un rapport d'évaluation est généré. Ce rapport contient une description des différents aspects de l'interface non conformes aux règles ergonomiques.

Destine permet l'ajout, la suppression et la modification de nouveaux critères grâce à un langage de définition de règles ergonomiques : le GDL (« *Guideline Definition Langage* ») (Beirekdar *et al.*, 2002). Ce langage structure les règles ergonomiques sous forme de fichiers XML. Il scrute le code HTML de la page pour dégager les valeurs des attributs à inspecter. Les résultats de l'évaluation sont fournis sous forme de pages web, Figure 2.5. L'évaluateur peut configurer ce rapport par critères ergonomiques, par objet, etc. Destine ne requiert que la présence d'un évaluateur connaissant HTML et l'ergonomie logicielle. Le processus d'évaluation est automatique. Ce système est flexible. En effet, il permet à l'évaluateur de personnaliser son évaluation suivant les guides, les critères ergonomiques, la priorité, l'objet, etc. L'usage du GDL par un utilisateur novice de Destine peut s'avérer fastidieux. De plus, l'usage de Destine pour l'aide à l'évaluation des *IHM* se restreint aux sites web.

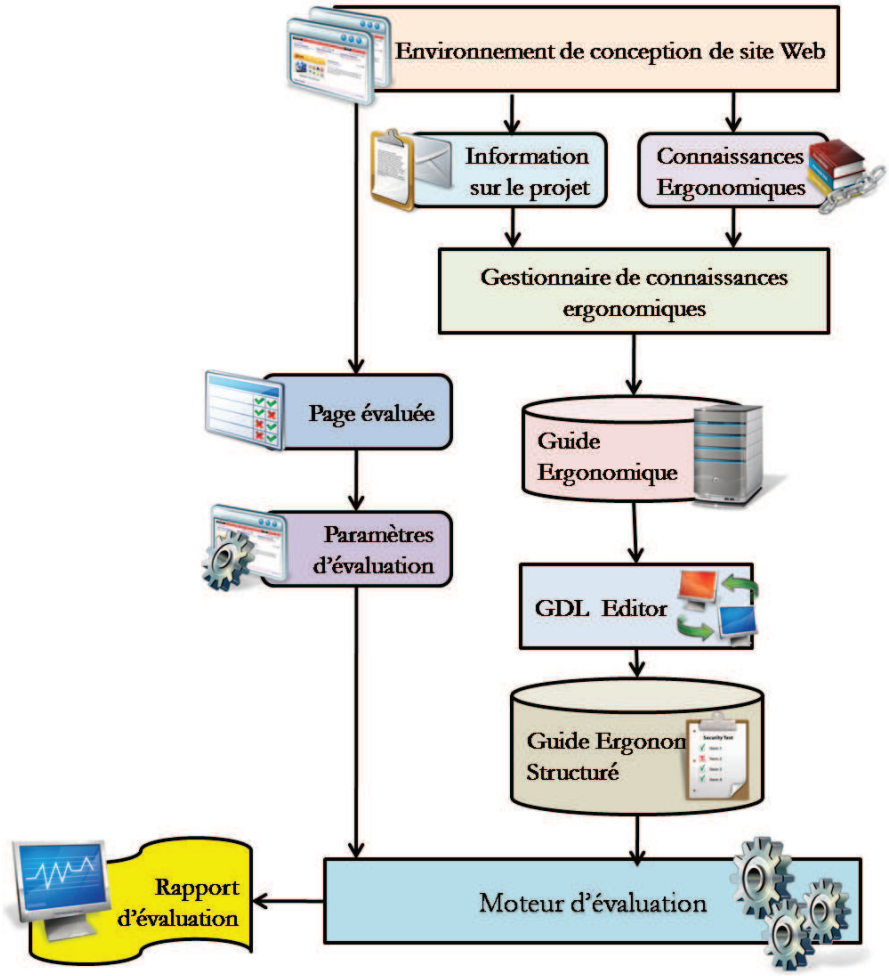



FIGURE 2.5: Architecture de Destine (Inspiré de (Beirekdar *et al.*, 2002))

#### 2.3.1.7 MAGENTA : Multi-Analysis of Guidelines by an Enhanced Tool for Accessibility (Leporini et Paterno, 2004; Leporini *et al.*, 2006)

Magenta est un outil d'évaluation statique de l'accessibilité dans des sites web via l'inspection des règles issues de différents guides ergonomiques. Ces guides sont codés dans des fichiers externes à l'application en XML et suivant un langage intitulé GAL (Guideline Abstraction Langage). Les données inspectées sont essentiellement les valeurs des attributs de la page à évaluer (exemple de la fonte d'écriture utilisée, nombre de couleurs utilisées, dimensions des composants graphiques utilisés pour la composition de l'interface-utilisateur, etc.). Pour certaines règles, qui relèvent de jugements qualitatifs plutôt que quantitatifs, l'évaluateur est amené à fournir des informations supplémentaires à l'outil pour assurer l'évaluation. A la fin du processus d'évaluation, Magenta fournit un rapport décrit en RDL [10]. Ce rapport regroupe les différentes incohérences à

10. RDL est un langage proposé par Microsoft dans le but de présenter les rapports sous forme standard. RDL se base essentiellement sur les fichiers XML pour les services de présentation des données. RDL peut intégrer des feuilles de calculs, des graphes, du texte et des images (Microsoft, 2010).

corriger dans la page évaluée. Un des avantages de Magenta est qu'il est facile à manier. En effet, l'évaluateur n'est pas censé bien maîtriser les langages de programmation (HTML, CSS, etc.). Magenta propose un éditeur graphique des guides ergonomiques : « *The Guideline Editor* ». Cet éditeur comporte une interface graphique pour éviter la manipulation des balises HTML. Néanmoins, le processus d'évaluation n'est pas totalement automatisé du fait que l'évaluateur doit parfois fournir au système des informations. Magenta ne traitent pas les informations pour l'évaluation ; il ne fournit qu'un rapport des recommandations non respectées dans l'interface évaluée.

#### 2.3.1.8 L'outil de Takata (Takata *et al.*, 2004)

L'outil proposé par Takata et ses collègues est un logiciel pour l'évaluation de l'accessibilité des pages web. L'évaluation se base sur des règles ergonomiques. Ces dernières sont formulées sous forme de fichiers XML spécifiés en SGSL (« *Simple Guideline Specification Language* »). Les auteurs du SGSL se sont inspirés du XPath [11]. La connaissance ergonomique n'est pas codée en dur dans le moteur d'évaluation. Cet outil permet l'ajout, la suppression et la modification de règles ergonomiques. L'outil propose une base fournie avec l'outil comportant une sélection de règles de *WCAG 1.0* (WCAG, 2011). Takata se base sur la transformation du code HTML et des règles exprimées en XSLT [12]. Dans le résultat de l'évaluation, cet outil fournit un fichier XML comportant l'ensemble de règles non respectées lors de la conception de l'interface graphique. Cet outil procède à une approche d'évaluation assimilée à une approche ensembliste ; il ne fournit pas le nombre de fois qu'une règle n'a pas été respectée ni les éléments relatifs à cette règle (Xiong, 2008).

#### 2.3.1.9 EvalAccess (Abascal *et al.*, 2004)

EvalAccess est un service Web destiné à l'évaluation de l'accessibilité des pages web. Il se base sur des travaux réalisés ultérieurement par Absacal : EvalIris (Abascal *et al.*, 2004). EvalAccess se base sur une collection de règles issues de *WCAG 1.0* (WCAG, 2011). Ces règles sont exprimées grâce à un langage de critères ergonomiques intitulé UGL (Unified Guidelines Language) (Arrue *et al.*, 2007a). Ainsi, les connaissances ergonomiques sont indépendantes du moteur d'évaluation. Afin de capture les informations concernant l'interface pour l'évaluation, EvalAccess parcourt le code HTML de l'interface à évaluer. Dans le résultat de l'évaluation cet outil propose un rapport d'évaluation sous forme de fichiers XML.

11. XPath est un langage qui permet d'effectuer des requêtes sur des fichiers XML (Williams, 2009)
12. XSLT est un langage de transformation XML. Il consiste à transformer un fichier XML en un document quelconque (Aman, 2009; Freire et de Mattos Fortes, 2005)

#### 2.3.1.10 WAEX : Web Accessibility Evaluator in a single Xslt file (Centeno *et al.*, 2006, 2007; Freire et de Mattos Fortes, 2005)

WAEX est un Service web pour l'évaluation de l'accessibilité des sites web. Il inspecte un certain nombre de règles ergonomiques. Ces règles sont stockées sous forme de fichiers XSLT. Ainsi les règles sont stockées en dehors du moteur d'évaluation et elles peuvent être ajoutées pour l'évaluation sans avoir à modifier l'outil d'évaluation. Afin d'évaluer une page web, WAEX parcourt les différents fichiers XML pour capturer les données de l'évaluation. Ensuite ces fichiers sont transformés en un seul fichier XML dans le but d'exploiter ces données pour l'évaluation. Il inspecte les recommandations ergonomiques du W3C/WAI [13] et celles du « *W3C's MobileOK Basic* » (W3C, 2008). En fin du processus d'évaluation, WAEX fournit à l'évaluateur un rapport de règles non respectées lors de la conception des pages web sous forme de page HTML. WAEX est un outil simple, léger, peu coûteux et facilement utilisable depuis le web. Il permet de n'inspecter que l'accessibilité des sites web et ne peut analyser que les fichiers XML. Le rapport d'évaluation porte essentiellement sur les problèmes d'accessibilité en lieu avec le site évalué.

### 2.3.2 Les outils codant en dur les connaissances ergonomiques

Dans la partie précédente, un ensemble d'outils a été présenté. Cet ensemble possède la spécificité de la possibilité d'alimenter l'outil d'évaluation par des règles ergonomiques sans modifier le code source de l'outil. En effet, la connaissance ergonomique n'est pas codée en dur dans l'outil d'évaluation. Dans cette partie, nous allons présenter les outils qui codent en dur la connaissance ergonomique dans leur source. Cet ensemble regroupe beaucoup plus d'outils que la catégorie précédente.

#### 2.3.2.1 CHIMES : Computer Human Interaction ModElS (Jiang *et al.*, 1992)

CHIMES est un outil qui permet d'évaluer automatiquement l'ergonomie d'une interface par rapport au guide d'OSF/MOTIF. Il est structuré autour de 4 modules :

- **Module d'acquisition des données.** Ce module permet d'acquérir les données de deux façons : automatique ou manuelle.
- **Module d'inspection de conformité.** Ce module convertit les données capturées par le premier module en informations d'entrée de la base de connaissances.

13. W3C/WAI est un ensemble de recommandations ergonomiques pour concevoir des sites web facilement accessibles (WCAG, 2011).

– **Base de connaissances.** Cette base contient les règles de l'évaluation. Il existe une centaine de règles ergonomiques.
– **Générateur de conseil.** Ce générateur fournit les résultats issus de l'évaluation et propose des améliorations de l'interface. Ces améliorations sont présentées sous forme de liste de conseils.

CHIMES inspecte l'utilisabilité des systèmes interactifs.

#### 2.3.2.2 KRI/AG : Knowledge-based Review of user Interfaces (Lowgren et Nordqvist, 1992)

KRI/AG permet d'évaluer les interfaces de type WIMP en inspectant leur utilisabilité. La capture des informations pour l'évaluation s'effectue via la conversion de la description de l'interface à évaluer en un fichier. Cette conversion se fait à l'aide d'UIMS[14]. KRI/AG est un système expert. Dans ce système, le fichier généré par UIMS sert de base de connaissances. La base des faits permet d'inspecter 70 règles ergonomiques. Ces règles proposées sont générales afin que l'outil puisse les interpréter convenablement (exemple : « *les items liés doivent être groupés du général au spécifique* »). En résultat d'évaluation, KRI/AG propose l'ensemble des incohérences ergonomiques détectées lors de l'évaluation de l'interface. Il n'offre pas de résultats plus élaborés à l'évaluateur. En effet, ce dernier doit assurer la phase de critique et de suggestion pour finaliser le processus d'évaluation. De ce fait, l'automatisation ne réside que dans la phase d'analyse. Les aspects couverts lors de l'évaluation s'articulent essentiellement autour de la présentation et l'affichage des informations sur l'interface. Le principal avantage de cet outil réside dans le fait qu'il est « *interactif* » lors du processus de l'évaluation. Il propose un ensemble de messages et de conclusions pour guider l'évaluateur tout au long du processus d'évaluation.

#### 2.3.2.3 Outil d'évaluation des menus de GIBBONS (Gibbons, 1992)

Gibbon propose un outil pour l'évaluation des systèmes interactifs. Cet outil se base sur un ensemble de 400 règles ergonomiques. Il n'évalue que les menus dans les interfaces graphiques. Afin de recueillir les données pour l'évaluation, Gibbons se base sur un questionnaire. Ce questionnaire est dédié à tous types d'utilisateurs (novice, expérimenté, etc.). Il couvre l'ensemble de tâches que peuvent subir les menus telles que recherche rapide, instructions habituelles, etc.). Cet outil décompte le menu suivant les différents aspects (par objet, par séquence d'actions élémentaires, etc.).

14. UIMS : User Interface Management System : un système pour la description des interfaces utilisateur (Olsen et Dan, 1991).

#### 2.3.2.4 Weblint (Bowers, 1996)

Weblint permet d'évaluer une interface web suivant plusieurs niveaux d'analyse : analyse lexicale (erreurs de balisage), analyse syntaxique (erreurs de balises), analyse structurelle, analyse fonctionnelle (absence de généricité dans certaines balises) et analyse stylistique (grammaire et conjugaison). L'ensemble des critères et des règles inspectées sont codés en dur dans l'outil. L'évaluateur ne peut pas ajouter ou modifier des règles. Cependant il peut sélectionner un ensemble de règles qu'il veut évaluer. En fin du processus d'évaluation, Weblint fournit un document texte regroupant l'ensemble des erreurs détectées.

#### 2.3.2.5 SHERLOCK (Mahajan et Shneiderman, 1997)

SHERLOCK [15] est un outil logiciel comprenant une suite d'outils pour l'évaluation des interfaces de type WIMP. Il permet d'inspecter l'utilisabilité des fenêtres de dialogue avec l'utilisateur. Afin de procéder à l'évaluation, tout d'abord la source de l'interface à évaluer est convertie sous une forme canonique qui est compréhensible et exploitable par l'outil SHERLOCK. Cette conversion est établie à l'aide d'un outil intitulé « Translator ». Notons qu'à chaque environnement de programmation est associé un « Translator ». SHERLOCK procède à un ensemble de tests de l'interface utilisateur à travers ses sept outils :

1. « The Dialog Box » : permet de lister les différentes boîtes de dialogue ainsi qu'un nombre de leur attribut tel que le nombre de composants par boîte, l'alignement des composants, etc.
2. « Margin Analyzer » : il s'agit d'une extension du premier outil, cet outil s'intéresse aux marges de l'interface évaluée ainsi que ses composants pour vérifier certaines propriétés telle que l'espacement entre les différentes composants de l'interface graphique.
3. « Interface Concordance » : permet d'extraire le contenu des composants graphiques tels que les boutons, les champs de texte et les libellés à partir de la source de l'interface à évaluer.
4. « Button Concordance » : permet de vérifier certaines propriétés des boutons de l'interface telles que leur taille, leur emplacement, les couleurs utilisées, etc. Ce module permet d'afficher la liste des boutons ainsi que leurs attributs suivant les boîtes de dialogue.
5. « Interface Spell checker » : permet de vérifier les fautes de saisie dans les interfaces et de dégager les appellations pouvant créer une confusion auprès de l'utilisateur.

15. Il est à noter que dans cet état de l'art, ils existent deux outils différents ayant le même nom : « Sherlock » proposés par deux auteurs différents (Mahajan et Shneiderman, 1997) et (Grammenos *et al.*, 2000)

6. « Terminology Baskets » : permet d'éviter d'utiliser différents synonymes pour la même action et ainsi d'éviter une confusion auprès de l'utilisateur (i.e. au lieu d'utiliser différents boutons ayant pour libellé « Cancel », « Delete » et « Clear », ces boutons auront le même libellé : « Cancel » à titre d'exemple).
7. « Button Layout Table » : permet de vérifier si une série de bouton est toujours affichée de la même façon de point vue espacement, ordonnancement, apparence, etc. Un exemple de série de bouton : les boutons « Ok », « Annuler » et « Aide ».

L'évaluation est automatisée lors de la capture et l'analyse des données pour l'évaluation. La phase de critique quant à elle est établie manuellement par l'évaluateur. Comme résultat d'évaluation, SHERLOCK propose un ensemble de données concernant l'interface, de statistiques et de comparaison à des recommandations pour la conception d'interfaces graphiques. Les recommandations intégrées dans l'outil sont proposées par des chercheurs à l'Université de Maryland (USA). L'outil proposé ne permet pas de modifier ces recommandations sans toucher à la source de SHERLOCK.

#### 2.3.2.6 HyperAT : Hyperpertext Authoring Tool (Theng et Marsden, 1998)

HyperAT permet d'évaluer l'utilisabilité des pages web grâce à une évaluation structurelle des pages et à des tests expérimentaux avec des utilisateurs. Les outils faisant appels à des utilisateurs pour l'évaluation d'utilisabilité sont assez rares. L'analyse structurelle consiste à inspecter certaines mesures telles que la profondeur de pages, le nombre de liens, etc. Quant à l'évaluation avec les utilisateurs, celle-ci consiste à analyser les données contenues dans un fichier log. Ce fichier est stocké sur le serveur. Il permet de donner certaines informations sur la fréquence de consultation d'une page, la fréquence de consultation de lien, etc. Ces informations, une fois analysées, permettent de générer, en fin de processus d'évaluation, un rapport pour l'aide à l'évaluation de l'interface utilisateur.

#### 2.3.2.7 Bobby (Cooper, 1999)

Bobby est un logiciel qui parcourt le code source HTML d'une page afin de détecter d'éventuelles incohérences ergonomiques. Cet outil inspecte un ensemble de 91 règles ergonomiques. Ces règles sont issues du WAI du W3C ou de la « *Section 508 guidelines* »[16]. Il est possible de sélectionner les règles ergonomiques nécessaires à l'évaluation. A chaque règle est assigné un indice de priorité. Cependant les résultats, donnés

16. La section 508 Standards est une loi américaine, qui porte sur l'accessibilité aux personnes handicapées des sites fédéraux et ressources électroniques du gouvernement (Act, 2012).

sous forme descriptive et graphique, sont assez fastidieux à interpréter, selon (Mariage et Vanderdonckt, 2005). De plus, Bobby évalue un site web essentiellement page par page.

#### 2.3.2.8 Dr Watson (Watson, 2011)

Dr Watson permet d'analyser les pages web dans le but d'évaluer leur qualité ergonomique. Il consiste en une batterie de tests pour évaluer l'utilisabilité d'un site web, ces tests sont :

- une analyse syntaxique du code HTML ;
- une vérification des liens brisés ;
- une analyse linguistique du texte contenu dans la page ;
- une estimation du temps de téléchargement de la page ;
- une analyse du nombre de mots contenus dans la page ;
- une analyse de la visibilité par rapport aux moteurs de recherche et,
- une vérification de la popularité de la page : nombre de pages ayant un lien vers la page évaluée.

Le rapport fourni en fin d'évaluation est une page HTML.

#### 2.3.2.9 Web Tango : Tool for Assessing NaviGation and Organization (Ivory et Hearst, 2001)

Web Tango est un outil destiné aux développeurs non professionnels afin d'élaborer des sites web de bonne qualité ergonomique. Web Tango évalue les pages web en élaborant des mesures quantitatives. Il se base sur une classification des sites web en trois profils pour les appliquer sur le système interactif. Web Tango est capable de comparer un site web à d'autres sites de bonne qualité ergonomique. Cette méthode permet de garantir : coût, rapidité d'exécution et objectivité du rapport d'analyse (Mariage et Vanderdonckt, 2005). Selon la vérification de la conception proposée par Web Tango elle ne remplace pas le test d'utilisabilité mais cette vérification et le test d'utilisabilité se complètent.

#### 2.3.2.10 ReWeb et TestWeb (Ricca et Tonella, 2000)

ReWeb et TestWeb sont deux outils logiciels autonomes mais qui se complètent dans le but d'évaluer des pages web grâce à un ensemble de tests renseignant sur la qualité fonctionnelle et structurelle des sites web. ReWeb et TestWeb se basent sur l'analyse des attributs graphiques et des interactions pour une évaluation de l'utilisabilité. A partir de la page à évaluer, ReWeb compile un modèle UML de cette page. Ensuite il procède à un ensemble d'analyses statiques. TestWeb teste le modèle généré par le premier module.

TestWeb se base sur la technique de White Box Testing. Ce couple d'outils permet une évaluation semi-automatique en se basant à la fois sur des données statiques et dynamiques. L'ensemble de règles évaluées sont codées en dur ainsi que l'ensemble des incohérences détectables par ces outils. Les tests intégrés par l'outil ne nécessitent pas de connaissances spécifiques pour le processus de l'évaluation.

#### 2.3.2.11 A-Prompt (A-Prompt, 2011; ATRC, 2002)

A-Prompt est un outil sous forme de service web accessible depuis le net (A-Prompt, 2011; ATRC, 2002). Il permet d'inspecter l'accessibilité d'un site web en se basant sur les règles du *WCAG 1.0* (WCAG, 2011), *WCAG 2.0*, la *section 508* et de l'*ItalianStancaAct*[17]. Cet outil parcourt le code HTML et même le code XHTML d'une page pour dégager les différentes valeurs d'attributs graphiques et en déduire les règles ergonomiques non respectées dans la conception des interfaces évaluées. Lors du processus d'évaluation, l'évaluateur peut restreindre le processus d'évaluation en certaines règles proposées dans l'outil. Bien que l'évaluateur dispose d'un grand choix en matière de règles ergonomiques, il ne peut pas ajouter des règles pour l'évaluation d'un site web quelconque. Le processus d'évaluation peut être automatique ou bien semi-automatique suivant la nature des règles évaluées. A-Prompt propose aussi des réparations automatiques pour certaines règles et des suggestions automatisées pour d'autres règles. Il fournit aussi une liste de règles dont il est impossible d'inspecter la conformité dans l'*IHM*. Le rapport d'évaluation, fourni à l'évaluateur, est sous forme de pages web contenant les aspects non respectés par rapport aux règles ergonomiques sélectionnées, des suggestions d'amélioration et les critères à vérifier manuellement. Le rapport est généré aussi en EARL.

#### 2.3.2.12 AccessEnable (Brinck et Hofer, 2002)

AccessEnable est un service web qui inspecte l'accessibilité des interfaces web en se basant que sur des règles issues du *WCAG 1.0* (WCAG, 2011) et de la Section 508. Le processus d'évaluation commence par la collecte des informations relatives au site à évaluer grâce à un robot logiciel : celui-ci est capable de parcourir les pages web pour stocker les différentes informations dans une base de données pour l'évaluation. Ensuite, il inspecte les différentes règles ergonomiques sélectionnées pour l'évaluation. Le résultat d'évaluation consiste en l'ensemble des incohérences ergonomiques détectées. Les règles non respectées sont marquées sous forme de tags sur le code source de l'interface évaluée.

17. Stanca Act est une loi italienne qui vise à favoriser l'accès des personnes handicapées aux TIC. Cette loi s'appuie sur les standards européens et internationaux.

Cet outil offre deux modes de correction : assistée et automatique paramétrable par l'utilisateur.

#### 2.3.2.13 WebXACT (Watchfire, 2010)

WebXACT est un service web qui évalue l'accessibilité des pages web suite à une analyse statique du code HTML. WebXACT est une continuité de deux outils existants conçus par le même éditeur : Web XML et WebQA. Il comprend un ensemble de règles issues notamment de *WCAG 1.0* (WCAG, 2011). Ces règles sont codées en dur dans l'outil. Elles portent essentiellement sur la qualité (contenu défectueux, informations sur la navigation, temps de chargement de la page, etc.). En fin d'évaluation, WebXACT fournit un rapport en HyperText des règles non respectées. Il ne permet ni de correction automatique ni semi-automatique de l'interface par rapport à ces règles.

#### 2.3.2.14 HTML Toolbox (NetMechanic, 2011)

HTML Toolbox permet d'inspecter l'utilisabilité d'une application web tout en parcourant et analysant son code HTML. HTML Toolbox existe en deux versions, l'une gratuite et l'autre payante. Il effectue des analyses lexicales, des analyses de la structure, des analyses syntaxiques, des analyses linguistiques et des vérifications des propriétés physiques. HTML Toolbox (dans sa version payante) permet la réparation automatique du site pour certaines règles ergonomiques. En fin de processus d'évaluation, HTML Toolbox fournit un rapport d'évaluation et la liste des aspects réparés sous forme HTML.

#### 2.3.2.15 LIFT Machine (Group, 2011)

Lift Machine est un outil d'évaluation de l'accessibilité et l'utilisabilité des *IHM*. Il permet d'évaluer des *IHM* des applications pour mobiles : interfaces *PDA* (Personal Digital Assistant) et appareils téléphoniques BlackBerry. Il évalue aussi les pages web. Cet outil s'intègre dans un environnement d'outils : Lift Mobile, Lift Text Transcoder (permet de compiler un contenu sous format HTML en format textuel), Lift for Macro media Dreamweaver (plug-in s'installant sous Dreamweaver), Lift Online (version en ligne de Lift Machine), etc. Lift Machine permet l'inspection de la validité d'un ensemble de 140 règles codées sur un serveur accessible depuis le réseau internet. Ces règles sont issues essentiellement des travaux de *WCAG 1.0* (WCAG, 2011) et de la Section 508. Lift machine offre une flexibilité pour la sélection des règles pour l'évaluation d'une *IHM*. Le processus d'évaluation est automatisé. Le service offert par l'outil est un rapport d'évaluation paramétrable constitué des aspects non conformes aux règles lors de l'inspection.

L'évaluateur peut effectuer à la fin de l'évaluation une réparation automatique des *IHM* pour les règles non respectées lors de la conception de cette *IHM*.

#### 2.3.2.16 WebSAT : Web StatiqueAnalyzer Tool (NIST, 2011)

WebSAT est un outil d'inspection de l'utilisabilité et de l'accessibilité des sites web. Il procède comme la majorité des outils d'évaluation des *IHM*, par le parcours du code HTML de la page à évaluer. Il comprend un ensemble de règles issues de deux corpus : les normes *ISO* (ISO, 1998, 1999) et une sélection de règles établie par l'éditeur de cet outil. La connaissance ergonomique est codée en dur dans l'outil d'évaluation. Cet outil distingue cinq catégories de règles : accessibilité, performance, maintenabilité, navigation et lisibilité. Cet outil va donc plus loin en termes de critères que l'ergonomie logicielle. Le rapport d'évaluation est constitué de l'ensemble des règles non respectées dans l'interface et de leur localisation dans le code source.

#### 2.3.2.17 Google Analytics (Google, 2011)

Google Analytics est un service proposé par Google dans le but d'analyser des sites web. Ce service d'analyse est gratuit ; en contrepartie l'utilisateur doit activer un autre service publicitaire de Google (« *Google AdSence* ») dans le site à évaluer. Google Analytics permet essentiellement l'analyse du trafic web. Une batterie de tests effectués par Google Analytics consiste essentiellement à améliorer les performances et l'accessibilité des sites web. Certains d'entre eux sont effectués par des robots pour tester le temps de téléchargement de la page.

Google Analytics s'intéresse aussi au contenu des sites web : cohérences entre méta balises et le contenu, cohérence entre contenu et descriptif, etc. De plus, il signale l'ensemble de liens et de fichiers pouvant constituer un risque pour les internautes. Cet outil propose un rapport détaillé des problèmes liés essentiellement à l'accessibilité du site évalué. Cependant, chaque fois que cet outil détecte une anomalie, il pénalise le site en le défavorisant dans le référencement des sites web. Cet outil propose des suggestions pour les réparations et l'amélioration du site essentiellement pour l'accessibilité et les performances (Google, 2011).

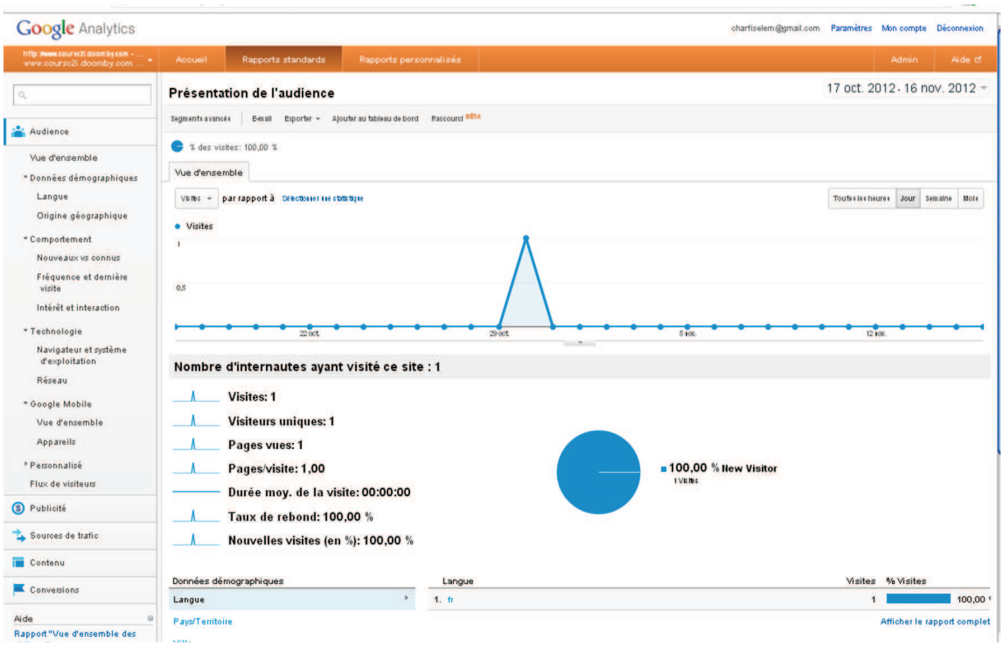



FIGURE 2.6: Google Analytics

#### 2.3.2.18 AWebHUT : Automated Web Home-Page Usability Tester (Rukshan et Baravalle, 2011)

AwebHUT est un outil qui permet d'inspecter un ensemble d'aspects relatifs à l'ergonomie des sites web. Ces aspects portent essentiellement sur : le nom de la page, la présentation du texte (écriture en gras, italique, etc.), la fonte d'écriture, la taille du texte, les spécifications des liens url, le contenu de la page, les messages d'erreurs, les images et les frames), etc. Le processus d'évaluation proposé est totalement automatisé. Il consiste à parcourir le code HTML de la page évaluée et faire appel aux différentes procédures stockées au niveau du code source de l'outil d'évaluation des sites web. A chaque procédure correspond une règle ergonomique. L'appel aux procédures se fait d'une façon séquentielle. En fin d'évaluation, cet outil propose deux rapports d'évaluation ; le premier est un résumé du processus d'évaluation et le second est un rapport détaillé. Les auteurs de cet outil, distinguent dans le résultat quatre niveaux de réponses pour chaque règle ergonomique : N (Neutre), V (Violée), R (Respectée) et E (Erreur). Le point fort de cet outil réside dans le fait qu'il contient une base de données des différents sites web inspectés. Il permet d'afficher les résultats d'évaluation des différents sites web. Ceci peut être très utile quand nous comparons diverses alternatives de conception.

#### 2.3.2.19 ErgoCoIn (Morandini *et al.*, 2011)

ErgoCoIn est un outil pour l'évaluation de l'utilisabilité dans les sites web et plus particulièrement les sites relevant de la catégorie de l'e-commerce. Il permet d'identifier et

de diagnostiquer les problèmes d'utilisabilité. Son usage est destiné à l'évaluateur non expert en ergonomie. ErgoCoIn mixe deux méthodes d'évaluation de l'utilisabilité : le questionnaire et l'inspection de l'utilisabilité. Ces deux méthodes, dans cet environnement, ne requièrent pas la participation directe des utilisateurs. Ceci permet de réduire le coût et la complexité du processus de l'évaluation. L'un des avantages de cet environnement est le fait qu'il propose en fin d'évaluation des rapports d'évaluation quantitatifs et subjectifs. Ce rapport est le résultat de la comparaison entre les informations émises d'une part par l'évaluateur et celles émises par l'utilisateur. Cette comparaison donne des indications sur les erreurs de conception dans l'interface et par conséquent les aspects à retravailler dans l'interface. D'après ces auteurs, ErgoCoIn est fiable, contextuel, objectif, systématique et peu coûteux (au sens du temps et de l'effort) (Morandini *et al.*, 2011).

## 2.4 Synthèse et discussion

Le degré de facilité de l'inspection de la conformité d'une interface par rapport à une règle ergonomique varie d'une catégorie de règle à une autre. Certaines règles recouvrent les aspects de présentation tels que la fonte d'écriture utilisée, le nombre de couleurs dans l'interface, etc. Leur inspection est assez facile à mettre en oeuvre dans des outils automatisés vu la facilité de l'extraction de l'information. Cependant, ces règles ne recouvrent que des problèmes mineurs ou encore cosmétiques dans l'interface à inspecter. D'autres règles sont plus difficiles à implémenter dans des outils automatisés pour l'évaluation des systèmes interactifs. Ces règles portent généralement sur des jugements qualitatifs et requièrent des métriques pour inspecter leur conformité.

Lors de l'évaluation ergonomique, la majorité des outils vérifient d'une façon automatisée la conformité de l'interface utilisateur par rapport à un ensemble de règles. Mais rares sont les outils qui permettent d'analyser et d'interpréter les résultats de telles vérifications d'une façon automatisée. La majorité des outils d'évaluation de la qualité ergonomique ne fournissent pas d'interprétation à l'évaluateur, ils se contentent de fournir un rapport de non respect des règles ergonomiques. Seulement 17 outils sur 28 outils présentés dans ce chapitre fournissent à l'utilisateur des interprétations automatiques, (cf. Annexe C : Tableau C.1, Tableau C.2 et Tableau C.3).

Lors de l'étude des outils existants, nous avons constaté que l'automatisation des outils se restreint généralement à la phase de capture des données. Quant à l'analyse et la critique de ces données, elles se font généralement par l'évaluateur du système. De plus, les résultats de l'évaluation proposés par ces outils s'avèrent peu pertinents. Certaines difficultés d'exploitation sont aussi relevées. En effet, les résultats d'évaluation consistent

généralement en l'ensemble des règles ergonomiques non respectées dans l'interface. De plus, la majorité des outils est destinée à l'évaluation des interfaces web. Il est très intéressent de concevoir des outils proposant une évaluation automatisée. L'automatisation permet essentiellement de réduire le coût de l'évaluation des systèmes interactifs. De plus, elle permet de couvrir plus d'erreurs de conception dans l'interface.

Nous constatons aussi que la majorité de ces outils se basent sur des données statistiques et ne prennent pas en considération l'utilisateur ainsi que l'utilisation du système. Les données dynamiques de l'interaction ne sont pas prises en compte (temps de réponse, exécution des tâches, séquence de tâches, etc.). Le recours à l'utilisateur lors du processus d'évaluation permet de dégager un grand nombre de problèmes d'utilité, d'utilisabilité et d'accessibilité. Il est préférable de procéder à des tests utilisateurs avec les utilisateurs finaux ou encore avec un échantillonnage de la population d'utilisateurs. Parmi les outils présentés précédemment, seulement un outil sur vingt analyse à la fois les données statiques et dynamiques. En effet, la majorité ne traite que les données statiques (25 outils parmi 28).

Les tableaux (Tableau C.1, Tableau C.2 et Tableau C.3) illustrent qu'une grande partie des outils d'évaluation ergonomiques ne s'applique qu'à l'évaluation des pages web. Il est bien commode d'évaluer ces pages du fait que toutes les données (liées à l'affichage et la présentation des informations) sont généralement codées dans le code source HTML de la page évaluée. A l'inverse des pages web, l'évaluation de la qualité ergonomique des applications logicielles est plus pénible vu la multiplicité des langages de programmation des logiciels (*C, C++, C#, Java, VB*, etc.). Ceci requiert des connaissances additionnelles de l'évaluateur en génie logiciel et en programmation logicielle. De plus, il est facile d'accéder au code HTML de n'importe quelle page et de le parcourir. Par contre pour les applications, le code source ne peut être exploré que s'il est fourni à l'évaluateur. Un quart (sept outils parmi 28) des outils présentés dans ce chapitre est destiné à l'évaluation des interfaces WIMP tandis que le reste vise les interfaces web. Dans les outils présentés, nous ne trouvons seulement que deux outils permettant l'évaluation des applications mobiles (Téléphone et PDA) (Billi *et al.*, 2010).

De plus, nous constatons clairement que l'expérience des utilisateurs n'est pas prise en considération pour l'évaluation de la qualité ergonomique de l'*IHM* (Law et van Schaik, 2010). La plupart des outils d'aide à l'évaluation des systèmes interactifs ne font pas appel à l'utilisateur durant le processus d'évaluation. Pourtant les résultats de l'évaluation seront meilleurs si l'utilisateur est impliqué dans le processus (Dray et Siegel, 2004; Schmettow, 2008). Il est souvent très utile d'impliquer la personne, censée se servir du système interactif, lors du processus d'évaluation. Pourtant l'utilisateur n'intervient pas dans le processus d'évaluation dans la majorité des outils présentés.

Les données requises pour l'évaluation des *IHM* des systèmes interactifs différent suivant les règles ergonomiques à inspecter. En effet certaines règles nécessitent uniquement des données sur des attributs de l'interface (Beirekdar, 2004; Gibbons, 1992), d'autres requièrent des informations sur l'interaction entre l'utilisateur et l'interface (Balbo, 1996). Certaines données sont assez difficiles voire même impossibles à détecter d'une façon automatique ; tel est le cas des critères relatifs à la clarté des messages d'erreur. Il semble plus intéressant de recourir à plusieurs méthodes de recueil de données du fait que chaque méthode extrait un ensemble de données. Nielsen confirme que les experts de l'utilisabilité recommandent l'usage de différentes techniques à la fois (Nielsen, 1993). Pour collecter ces données, les évaluateurs utilisent généralement des mesures de performances et les complètent par des mesures de préférences (entretien, questionnaire, etc.) (Charfi *et al.*, 2009).

Il est clair que l'évaluation de la qualité ergonomique possède des avantages. Nous pouvons citer essentiellement la facilité de détection de problèmes ergonomiques majeurs. De plus, généralement cette évaluation ne requiert pas beaucoup de participants lors de l'évaluation. Par rapport aux autres méthodes (relevant des approches empiriques ou analytiques au sens de (Bastien et Scapin, 2001), l'évaluation de la qualité ergonomique est rapide à mettre en place et à effectuer. De plus, elle s'avère peu couteuse. Néanmoins, nous relevons certains inconvénients dans cette évaluation. Les inconvénients s'articulent essentiellement autour du fait que de telle évaluation ne peut être mise en place qu'avec le système interactif fini ou encore avec des prototypes avancés. De ce fait, l'évaluateur est restreint à l'usage de cette méthode à la fin du cycle de développement. De plus, ces outils n'indiquent pas le niveau global de la qualité ergonomique (excellente, moyenne, passable, catastrophique, etc.). Elles nous renseignent en général sur les incohérences ergonomiques. L'évaluation est plutôt d'ordre subjectif. En effet, le degré de la gravité des problèmes détectés relève d'un jugement qualitatif de la part de l'évaluateur. De plus, le problème au niveau de la subjectivité réside aussi au niveau de l'interprétation des problèmes détectés. En effet, ils varient d'un évaluateur à un autre.

## Conclusion

Le monde dans lequel nous vivons, ne cesse de croître au rythme des technologies. Les ordinateurs sont devenus des outils de travail indispensables dans bien de domaines. Ceci a entrainé l'apparition d'un nouveau défi visant l'obtention d'un meilleur coût unitaire, d'augmenter l'efficacité et la performance. Pour y arriver, de nombreux aspects dans le processus de développement des systèmes interactifs doivent être assurés. L'un des aspects les plus importants réside dans l'interface utilisateur du système. De ce fait,

l'évaluation de l'interface proposée à l'utilisateur s'avère primordiale. Ainsi beaucoup de méthodes et techniques ont été mises en place pour l'obtention d'interfaces accessibles, utiles et utilisables. Les travaux dans ce contexte sont abondants. Dans ce chapitre, nous nous sommes focalisés sur les outils d'évaluation des systèmes interactifs exploitant les connaissances ergonomiques. Nous avons présenté un ensemble représentatif d'outils logiciels pour l'évaluation des systèmes interactifs proposés par la communauté *IHM*. Ces outils différent suivant certains aspects : les interfaces qu'ils évaluent, le(s) facteur(s) de qualité inspecté(s), la nature et le type de l'évaluation, le degré d'automatisation du processus d'évaluation, etc.

Des classifications existent pour les méthodes d'évaluation des Interfaces Homme-Machine (par exemple (Ivory et Hearst, 2001)). Mais aucune n'est dédiée aux outils à base de règles pour l'évaluation ergonomique des interfaces utilisateur. Dans ce chapitre, nous avons proposé une classification de ces outils. La classification se base sur la flexibilité de l'outil vis-à-vis des règles ergonomiques inspectées. Notre classification identifie deux grandes catégories d'outils. La première catégorie regroupe l'ensemble des outils dont les règles ergonomiques sont codées en dur dans l'outil d'évaluation. Ils offrent la même évaluation à tous les systèmes interactifs. Ils ne permettent pas l'intégration de nouvelles règles pour le processus d'évaluation (et ni même la modification des règles existantes). C'est ce qu'on appelle le codage en dur de la connaissance ergonomique. La seconde catégorie concerne l'ensemble des outils offrant à l'évaluateur d'intégrer de nouvelles règles sans toucher au code source de l'outil d'évaluation. Généralement ces outils adoptent des langages de définition de règles ergonomiques. La définition consiste à exprimer ces règles sous forme formelle et exploitable pour le processus d'évaluation. Il est à rappeler que les règles sont souvent exprimées en langage naturel. De ce fait, ces outils et langages sont nécessaires pour la formalisation de règles afin qu'elles soient exploitables par les outils d'évaluation.

L'évaluation, basée sur l'inspection de la conformité de l'interface par rapport à un ensemble de règles ergonomiques, se distingue par rapport aux autres techniques d'évaluation par sa facilité de détection de problèmes ergonomiques majeurs. D'autre part, cette évaluation s'avère peu couteuse par rapport aux autres méthodes d'évaluation. Néanmoins, l'évaluation ne peut être appliquée que lors des phases finales ou bien avancées du cycle de développement des systèmes interactifs. De plus ces outils, ne précisent pas explicitement la qualité ergonomique finale. Ils se contentent de renseigner l'évaluateur sur les incohérences ergonomiques qu'ils peuvent détecter. De ce fait, l'évaluation semble ainsi relever de l'interprétation de l'évaluateur plutôt qu'une évaluation objective. Ce manque d'objectivité est du essentiellement au fait que les rapports d'évaluation relèvent souvent de jugements qualitatifs. De plus, l'utilisabilité d'un système interactif

donné ne dépend pas uniquement des aspects du système ; elle dépend aussi des caractéristiques de l'utilisateur, des tâches qu'ils exécutent et l'environnement de travail (du point de vue : technique, organisationnel et physique).

Chapitre 3

# Etat de l'art des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'interface et l'utilisateur

> « *Il y existe deux manières de concevoir un logiciel. La première, c'est de le faire si simple qu'il est évident qu'il ne présente aucun problème. La seconde, c'est de le faire si compliqué qu'il ne présente aucun problème évident. La première méthode est de loin la plus complexe.* »
>
> Tony Hoare

## Introduction

Le thème de l'évaluation des *IHM*, comme mentionné dans le premier chapitre, est un thème très riche en concepts, méthodes, techniques et travaux. Le second chapitre nous a permis d'avoir une idée sur certains outils d'évaluation ainsi que sur leurs limites. Ces outils ont la spécificité de se baser sur des travaux relatifs à l'ergonomie logicielles sans autant s'intéresser au noyau fonctionnel du système interactif à évaluer ou encore à l'interaction entre l'utilisateur et le système interactif. Ce présent chapitre s'intéresse à un autre ensemble d'outils. Il s'agit des outils permettant l'évaluation des systèmes interactifs tout en se basant sur l'interaction entre l'utilisateur et le système évalué dans le but d'identifier les problèmes d'utilité, d'utilisabilité et d'accessibilité des *IHM*.

Ce chapitre est structuré suivant trois sections. La première s'intéresse à introduire les aspects considérés lors de la présentation des outils. La seconde comporte un état de l'art des outils d'évaluation basés sur l'interaction entre l'utilisateur et le système évalué. Finalement, une troisième section est dédiée à une synthèse et une discussion des fonctionnalités et des limites des outils présentés.

## 3.1 Aspects considérés pour la présentation des outils d'évaluation

Lors de la présentation des outils évaluant les Interfaces Homme-Machine, nous avons opté pour présenter ces outils suivant un certain nombre d'aspects. Ces aspects sont :

- **Application.** Cet aspect vise à déterminer la nature des interfaces évaluées : WIMP, WUI ou Mobile UI.
- **Type d'outil.** L'outil d'évaluation peut être un web service, une application logicielle ou un plugin.
- **Principe d'évaluation adopté.** Comme mentionné dans le premier chapitre, il existe divers principes pour procéder à l'évaluation des interfaces utilisateurs. L'évaluation peut être établie via la confrontation entre deux séquences d'interaction ainsi que la capture et l'analyse du mouvement de la souris, la saisie du texte à partir du clavier.
- **Rapport d'évaluation fourni.** Le rapport d'évaluation peut comporter un ensemble de problèmes d'utilité et d'utilisabilité relevés lors de l'évaluation. Il peut aussi intégrer des informations sur la qualité de l'interface évaluée et même des suggestions de corrections et d'améliorations de l'interface.
- **Nature des événements capturés.** Les outils d'évaluation présentés dans ce chapitre se basent sur l'interaction entre l'utilisateur et le système évalué. Ils capturent et analysent essentiellement des événements de l'interaction. Dans cet aspect nous nous focalisons sur la nature de ces événements. Pour s'y faire, nous nous basons sur la classification proposée par Hilbert et Redmiles (Hilbert et Redmiles, 2000), Figure 1. Elle consiste à départager les différents événements suivant six niveaux. Le premier englobe le niveau le plus haut ; il s'agit du but des actions tel est le cas d'une réservation d'un billet d'avion. Ensuite, le second niveau concerne les tâches permettant d'atteindre le but de la séquence d'action tel est le cas de l'identification sur la plateforme, le choix du vol, le paiement, etc. Le troisième niveau implique l'ensemble des actions relatives à un même objet de l'interface graphique : le niveau de l'abstraction de l'interaction telles sont les actions relatives au champ de texte : adresse mail du client. Puis, le quatrième niveau regroupe les événements interface utilisateur comme l'appui sur des touches du clavier. Le cinquième niveau invoque les actions exécutées sur les périphériques d'entrée. Le dernier niveau, quant à lui, renseigne sur les actions physiques de

l'utilisateur (appui sur une touche quelconque, clic souris, etc.). Ce niveau indique les actions de bas niveau.

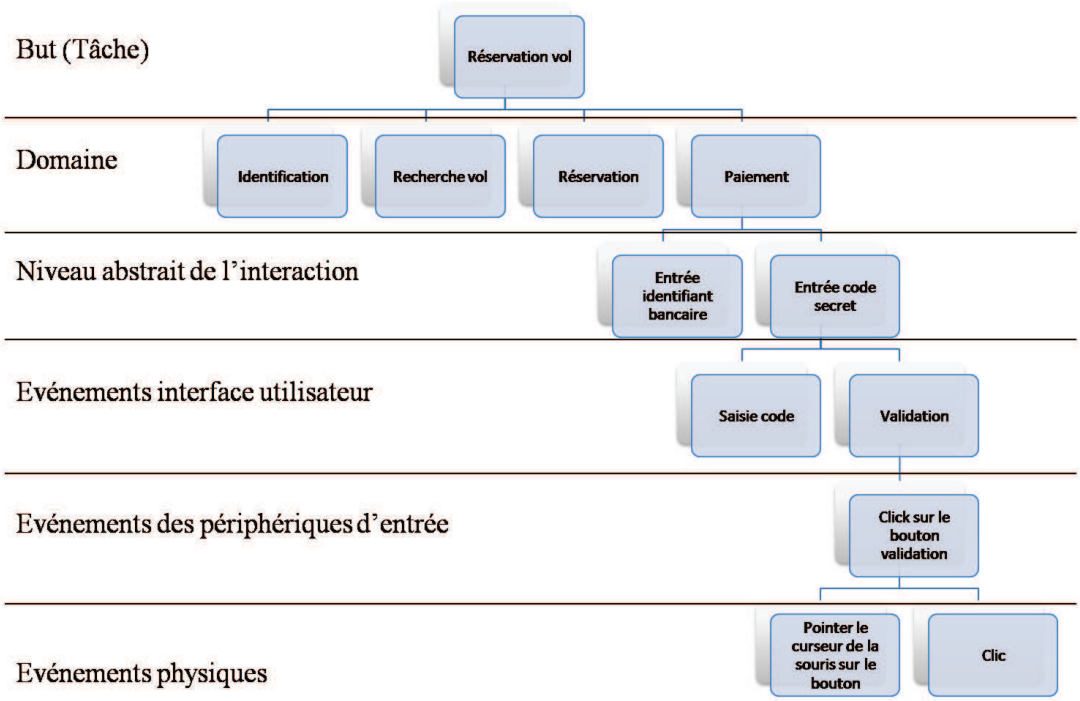



FIGURE 3.1: Extrait d'un exemple de décomposition d'une séquence d'action suivant l'arborescence proposée par Hilbert et Redmiles (Hilbert et Redmiles, 2000)

- **Les intervenants lors du processus d'évaluation.** Cet aspect introduit l'ensemble des opérateurs humains intervenant lors du processus d'évaluation ainsi que la tâche de chacun. Les intervenants peuvent être : utilisateur final, utilisateur expérimental, évaluateur et concepteur.
- **Phase d'application lors du cycle de développement des systèmes interactifs.** consiste à indiquer si l'outil fait partie des outils d'évaluation précoce ou encore de l'évaluation du système final (phase de test).
- **Automatisation de l'outil d'évaluation.** Consiste à indiquer le degré d'automatisation de l'outil présenté lors des trois phases capture, analyse et critique. Chaque phase peut être manuelle, semi-automatique ou automatique (Ivory et Hearst, 2001).
- **Fonctionnalités supportées.** Cet aspect vise à indiquer les services proposés par l'outil d'évaluation. Ces services peuvent être : identification des problèmes d'utilisation de l'interface utilisateur, la réparation (automatique ou assistée) de l'interface évaluée, des heat maps [1] du clic de la souris.

1. Une heat map est une carte représentative des points de fixations du regard ou encore du curseur de la souris. Généralement, les endroits les plus visés sont mentionnés en rouge. Ce genre de carte est généralement utilisé dans le thème de l'ergonomie logicielle.

– **Interactivité.** Elle porte sur la facilité d'utilisation de l'outil d'évaluation. L'utilisation peut être facilitée via le guidage proposé lors du processus d'évaluation ou encore la nature des interfaces de l'outil.
– **Dimensions traitées.** Cet aspect indique les dimensions de qualité pris en considération par l'outil lors de l'évaluation (l'utilisabilité, l'utilité et l'accessibilité).
– **Technique de capture.** Cet aspect s'intéresse à la technique adoptée pour la capture des actions utilisateur afin de les exploiter pour l'évaluation. La capture peut s'élaborer via l'utilisation d'un mouchard électronique (cf. Chapitre 1), l'exploitation de fichiers logs, l'observation de l'utilisateur interagissant avec l'interface graphique (observation directe ou bien en différée), l'observation et l'analyse des mouvements des yeux de l'utilisateur (Eye tracking) et le suivi et l'analyse du mouvement de la souris.

## 3.2 Les outils d'évaluation

Les outils d'évaluation sont présentés dans cette section suivant leur ordre chronologique d'apparition.

### 3.2.1 EMA (Balbo, 1996)

EMA est un acronyme pour « ***E****valuation par* ***M****écanisme* ***A****utomatique* ». Il consiste en un mouchard électronique pour la validation des règles ergonomiques dans les interfaces utilisateur de type WIMP. L'évaluateur procède tout d'abord à l'insertion de lignes de commande dans la source du système final à évaluer. Ensuite, il spécifie le modèle de la tâche. Puis, un (des) utilisateur(s) exécute(nt) cette tâche. Les actions capturées sont stockées sous forme de séquences. EMA permet de considérer les différents niveaux d'événement au sens (Hilbert et Redmiles, 2000). L'évaluation à l'aide d'EMA permet de détecter les ruptures, les annulations immédiates, les répétitions et l'invalidité dans la séquence de manipulation de l'interface. Néanmoins, les auteurs d'EMA se restreignent à un ensemble de critères inspectés par ce système. De plus, il n'est pas évident que l'évaluateur dispose du code de l'application à évaluer. En fin d'évaluation, EMA ne propose pas de rapport d'évaluation. Il se contente de proposer quelques conclusions portant sur la comparaison entre le modèle de tâche et le modèle utilisateur. EMA se base sur l'utilité et l'utilisabilité pour l'évaluation des interfaces utilisateurs.

### 3.2.2 USINE (Lecerof et Paterno, 1998)

USINE [2] est un acronyme pour « ***US****er* ***IN****terface* ***E****valuator* ». Il s'agit d'un outil logiciel pour évaluer les interfaces utilisateur de type WIMP tout en se focalisant sur l'utilisabilité de ces interfaces. Pour s'y faire, il se base sur l'analyse du fichier log. Ce dernier comporte les différentes actions élémentaires exécutées par l'utilisateur. Ces actions sont enregistrées dans le fichier log grâce à des outils logiciels que l'évaluateur doit installer au préalable : JDK [3] et QC/Replay [4]. QC/Replay sauvegarde les actions sous forme de fichiers log qu'USINE exploite pour l'évaluation. Cette dernière consiste à comparer les actions issues du fichier log à un modèle de tâche spécifié en CTT [5] par le concepteur. Il vise à indiquer les taux d'accomplissement des tâches prévues pour être exécutées par l'utilisateur (réussite, échec, retard, etc.). A la fin de l'évaluation, USINE propose à l'évaluateur un ensemble de données numériques concernant l'interaction (temps d'exécution, nombre de tâches exécutées, erreurs d'exécution, etc.). L'évaluation avec USINE ne peut avoir lieu que dans le cas où le système est finalisé ou bien dans le cas d'un prototype avancé.

2. USINE fait partie d'une famille d'outils développés par l'équipe « *The Human-Computer Interaction Group* » au sein du laboratoire HIIS Laboratory. Cette famille comporte aussi : Web RemUsine, MM RemUsine et Multi-device RemUsine.

3. Java Development Kit est un ensemble de bibliothèques logicielles pour les applications développées sous Java. `http://www.ibm.com/developerworks/java/jdk/`

4. QC/Replay est un outil utilisé généralement par les outils d'évaluation des interfaces graphiques. Il permet de capturer les actions exécutées par un utilisateur durant une session de travail. Il permet aussi de rejouer ces actions. `http://www.uniforum.org/`

5. CTT est une notation pour la modélisation des tâches utilisateurs. L'un de ses principaux avantages est la structuration hiérarchique des tâches. De plus, nous citons les opérateurs de modélisation qu'elle offre telles que les opérateurs temporels (Paterno *et al.*, 2007b, 1997). Un environnement est proposé pour supporter cette notation lors de la modélisation ; il s'agit de CTTE (HIIS, 2012).

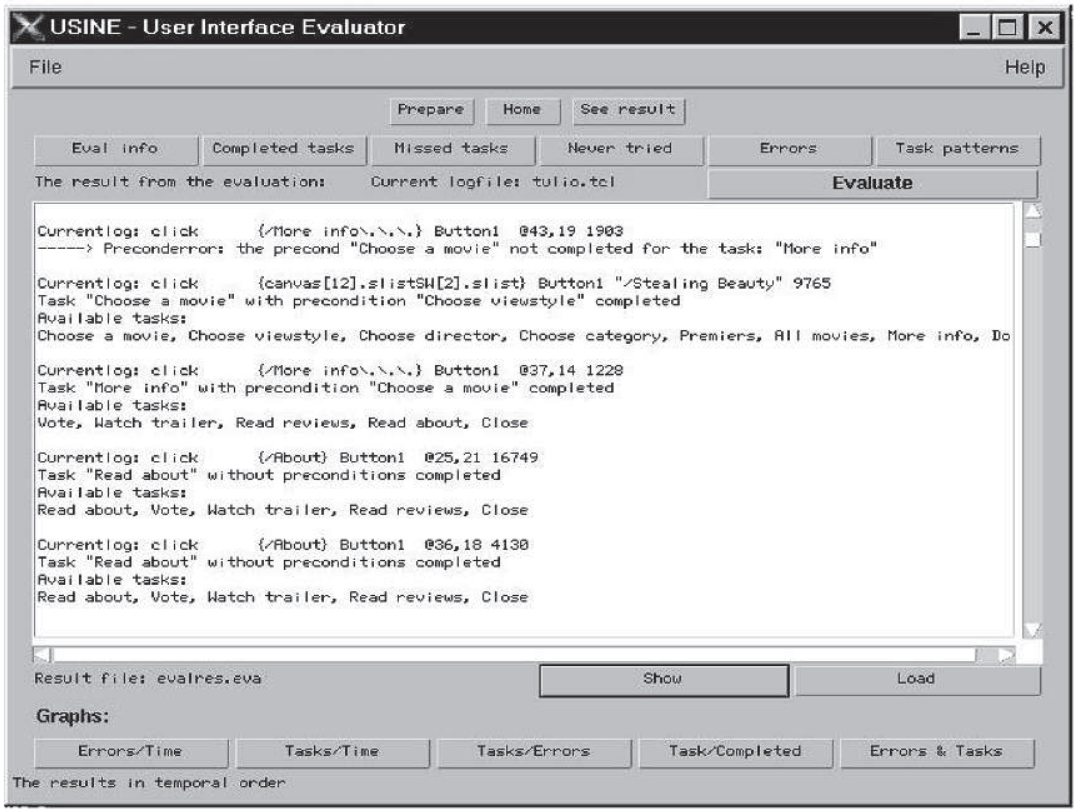



FIGURE 3.2: Exemple de résultats d'évaluation fournis par USINE (figure extraite de (Lecerof et Paterno, 1998))

### 3.2.3 RemUsine (Paterno et Ballardin, 2000)

RemUsine a un principe d'évaluation très similaire à USINE. La seule différence réside au niveau de la possibilité d'élaborer une évaluation à distance et avec plusieurs utilisateurs. Cette différence présente un inconvénient au niveau de l'évaluation. En effet, il est impossible d'avoir une idée sur le contexte d'utilisation à travers cet outil quand l'utilisateur exploite l'interface à distance. A titre d'exemple, lors de faits troublants (tel est le cas d'un appel téléphonique, une interruption quelconque, environnement bruyant, etc.), il serait bien difficile de bien exploiter et étudier les séquences d'actions, Figure 3.3. L'appellation RemUsine correspond à l'acronyme de « ***REM****ote* ***US****er* ***IN****terface* ***E****valuator* ».

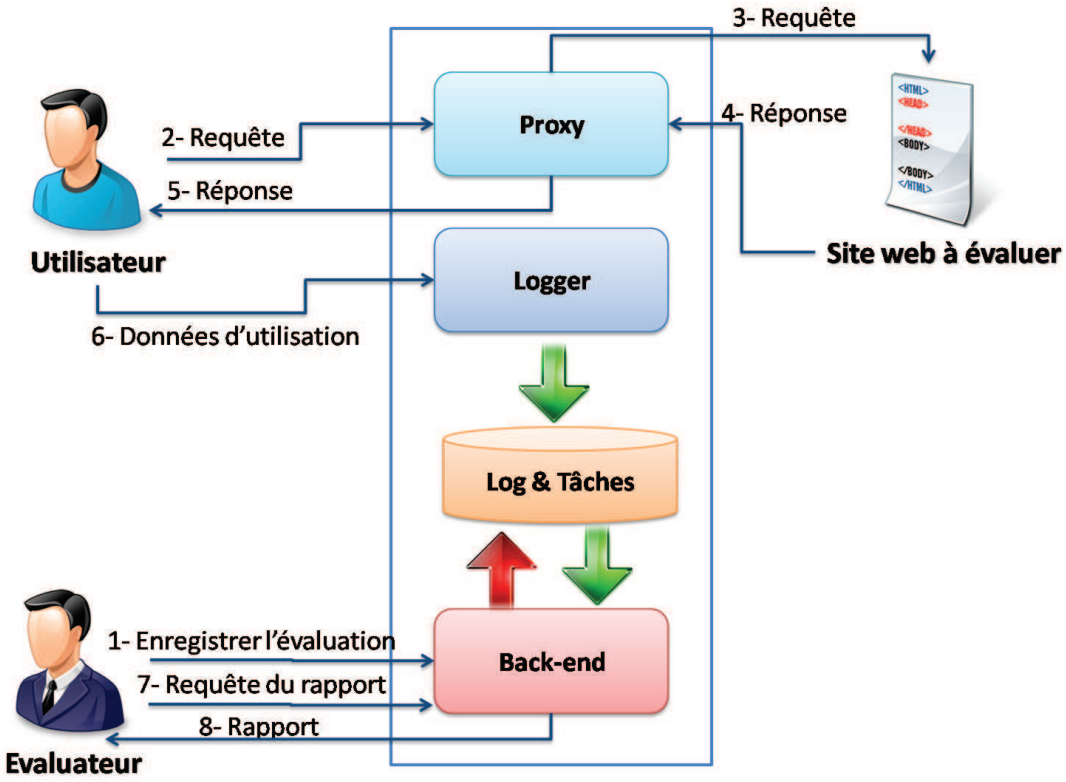



FIGURE 3.3: Architecture fonctionnelle de RemUsine (Carta *et al.*, 2011)

### 3.2.4 Wet (Etgen et Cantor, 1999)

Wet est un web service dédié à l'évaluation des interfaces web. Tout comme Web RemUsine. Wet permet de capturer les informations de niveau bas (événements physiques au sens de (Hilbert et Redmiles, 2000)). Comme résultat d'évaluation, Wet propose un ensemble d'informations portant essentiellement sur le chargement des pages ainsi sur la navigation entre elles de la part de l'utilisateur. Les auteurs de Wet ont opté pour le choix de l'« event handling » du navigateur internet dans le but de simplifier le processus d'évaluation en évitant l'évaluateur d'installer des outils supplémentaires. Les informations capturées par cet outil sont sauvegardées dans les fichiers cookies et ensuite transférées dans une base de données. Wet renseigne aussi l'évaluateur avec des données numériques portant par exemple sur le nombre des tâches accomplies (avec le taux de réussite) ainsi que le temps d'exécution. La nomenclature de cet outil est issue de l'acronyme pour « ***W**eb **E**vent-logging **T**ool* ». L'usage de Wet s'effectue lors de la dernière phase du cycle de conception des systèmes interactifs : la phase de test. Il vise essentiellement à inspecter l'utilisabilité des sites-web.

| Metrics | | Objectives and Benchmarks | Usability Participants' Results Compared to Benchmarks | Satisfy Objective? |
|---|---|---|---|---|
| **Task completion** | | | | |
| | Task 1 | | 100% completed task | yes |
| | Task 2 | | 100% completed task | yes |
| | Task 3 | | 75% completed task | yes |
| | Task 4 | 75% complete task for all tasks | 50% completed task | no |
| | Task 5 | | 100% completed task | yes |
| | Task 6 | | 75% completed task | yes |
| | Task 7 | | 100% completed task | yes |
| **Time to complete task*** | | | | |
| | Task 1 | 75% less than 20 seconds | 100% completed task in time | yes |
| | Task 2 | 75% less than 25 seconds | 75% completed task in time | yes |
| | Task 3 | 75% less than 45 seconds | 100% completed task in time** | no |
| | Task 4 | 75% less than 35 seconds | 0% completed task in time*** | no |
| | Task 5 | 75% less than 90 seconds | 75% completed task in time | yes |
| | Task 6 | 75% less than 30 seconds | 33% completed task in time** | no |
| | Task 7 | 75% less than 120 seconds | 75% completed task in time | yes |
| **Deviation from ideal model** | | | | |
| | Task 1 | | 100% had no more than 2 deviations | yes |
| | Task 2 | | 100% had no more than 2 deviations | yes |
| | Task 3 | | 75% had no more than 2 deviations | yes |
| | Task 4 | % have no more than 2 deviations for all tasks | 50% had no more than 2 deviations | no |
| | Task 5 | | 100% had no more than 2 deviations | yes |
| | Task 6 | | 75% had no more than 2 deviations | yes |
| | Task 7 | | 100% had no more than 2 deviations | yes |

FIGURE 3.4: Exemple de résultats d'évaluation proposé par Wet

### 3.2.5 iBot (Zettlemoyer *et al.*, 1999)

iBot est un outil logiciel pour l'évaluation des interfaces de type WIMP lors de la dernière phase du cycle de conception des systèmes interactifs. Il permet d'inspecter le critère d'utilisabilité par la détection des événements de bas niveau (événements physiques au sens de (Hilbert et Redmiles, 2000)). Il comporte un agent logiciel permettant de simuler le comportement d'un utilisateur vis à vis l'interface évaluée et de ré-visionner l'ensemble des actions élaborées par cet agent dans le but d'identifier les actions erronées, inutiles ou encore répétitives. Cet agent est constitué essentiellement de trois modules :

- **Module de traitement des images.** Ce module permet d'acquérir sous forme de captures d'écran ce que l'utilisateur est censé voir ;
- **Module de gestion des événements.** Ce module permet de capturer et de rejouer les séquences d'actions exécutées par l'agent et,
- **Module de représentation des états internes.** Ce module permet de combiner entre les sorties du premier et second module en associant les événements aux images et par conséquent d'identifier sur l'interface les problèmes détectés lors de l'interaction (actions inutiles, tâches inaccomplies, etc.).

L'évaluation avec iBot est automatisée lors de la capture et l'analyse. Par contre, la phase de critique est établie manuellement par l'évaluateur. L'appellation de cet outil correspond à « *interface soft**BOT**s*[6] ».

6. Softbot = Software Robot.

### 3.2.6 Web Quilt (Hong *et al.*, 2001; Waterson, May 2002)

Web Quilt est un outil dédié pour l'évaluation des interfaces WUI. Il a l'avantage d'être compatible avec les différents systèmes d'exploitation et des différents navigateurs. A la différence de nombreux outils présentés dans ce chapitre, Web Quilt ne requiert pas l'installation d'outil logiciel pour la détection et la capture des événements utilisateur sous formes de fichiers logs. Il se base sur un proxy pour la capture des actions utilisateurs (niveau abstrait de l'interaction au sens de (Hilbert et Redmiles, 2000)). Il se base sur trois modules pour l'évaluation des interfaces utilisateurs :

- « ***ProxyLogger*** ». Ce module permet de réceptionner les différentes requêtes de l'utilisateur et d'en extraire les données utiles pour l'évaluation. Ces données sont enregistrées sous forme de fichiers log. Ensuite, ce module transfère la requête au serveur et réceptionne la réponse qu'il transfère à l'utilisateur. Ainsi il sert d'interlocuteur entre le serveur et l'utilisateur. Il stocke aussi dans sa mémoire cache toutes les pages utilisées ;
- **Inféreur d'actions.** Ce module permet d'analyser et d'exploiter le fichier log issu du premier module et,
- **Module de fusion des graphes.** Ce module permet de fusionner entre les différents fichiers log pour en déduire un graphe. Ce dernier est constitué de noeuds et de flèches où les noeuds correspondent aux pages web et les flèches à la transition entre ces pages. Le graphe généré est bien présenté de façon que l'évaluateur puisse détecter les difficultés d'utilisation du site web. Les flèches sont colorées suivant la durée des transitions (où la couleur rouge correspond à une longue durée de visualisation et la couleur jaune correspond à une durée courte). De plus, la fréquence d'utilisation des transitions est indiqué via l'épaisseur des flèches (plus la transition est utilisée et plus elle est épaisse).

Ainsi cet outil renseigne en fin d'évaluation sur la fréquence de consultation des pages ainsi que les transitions entre ces pages. Cet outil n'est utilisable que lors de la dernière phase du cycle de conception des systèmes interactifs. Il vise à détecter les problèmes d'utilisabilité dans les sites web.

### 3.2.7 SMT (Speedzinemedia, 2012)

SMT est un web service pour l'évaluation des interfaces web. Il se base sur la capture et l'analyse du mouvement de la souris. Ce web service est structuré sous forme de solution client-serveur. De plus, il est open source. SMT se base sur des études qui affirment la corrélation entre le mouvement des yeux et le mouvement du pointeur de la souris (Mueller et Lockerd, 2001; Chen *et al.*, 2001; Pan *et al.*, 2004). Ainsi il permet de capturer et de rejouer le mouvement du curseur de la souris afin d'identifier les aspects du site sur

lesquels l'utilisateur s'est focalisé et les difficultés rencontrées lors de l'interaction avec l'interface utilisateur. L'interface proposée par les éditeurs est bien agréable à utiliser. De plus, le guidage est parfaitement assuré dans ce site ce qui implique un usage intuitif de la part d'utilisateurs novices. D'ailleurs les utilisateurs peuvent ajuster l'apparence du site suivant leurs préférences et besoins. Les données détectées portant sur le mouvement de la souris peuvent être exportées suivant plusieurs formats (XML, csv, etc.) ou encore dans une base de données de type « *mySQL* ». Le critère inspecté par cet outil est l'utilisabilité. Néanmoins SMT ne permet pas de s'entretenir avec les utilisateurs pour leur demander des justificatifs sur des actions qu'ils ont effectuées. La nomenclature de ce web-service provient de « ***S**imple **M**ouse **T**racking* ».

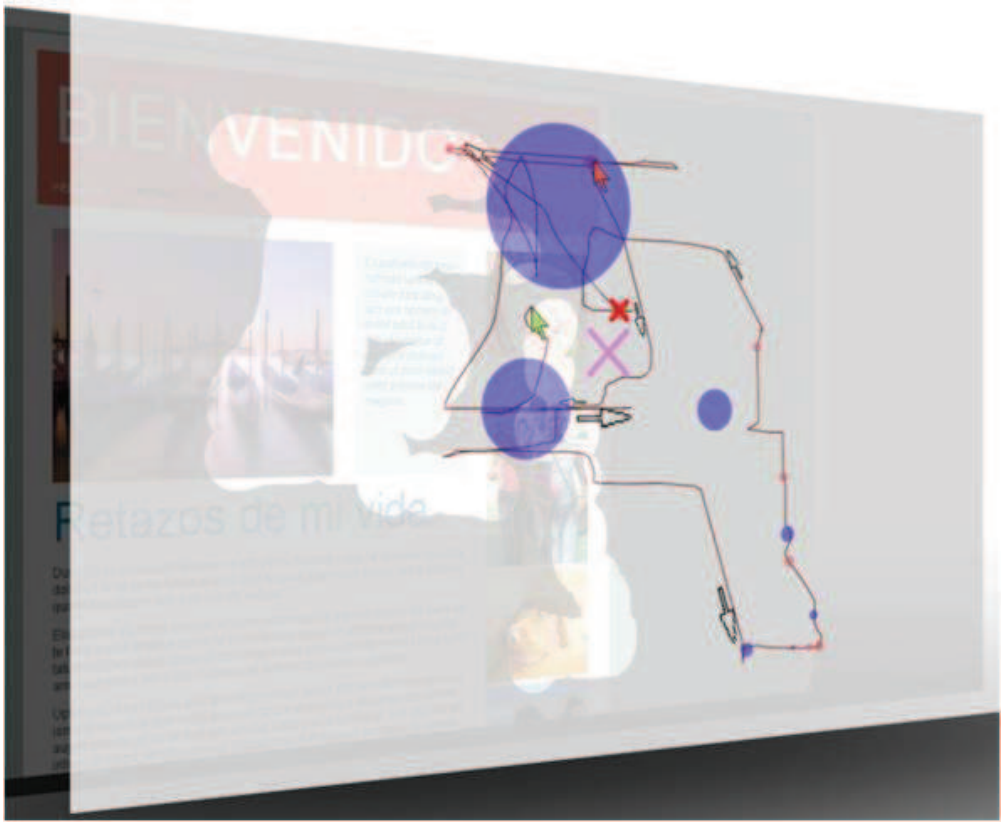


FIGURE 3.5: Un exemple d'une capture de mouvement de la souris avec SMT (Speedzinemedia, 2012)

### 3.2.8 WebRemUsine (Paganelli et Paterno, 2002, 2003)

WebRemUsine constitue une continuité de RemUsine. La différence réside au niveau du type des interfaces évaluées ; cet outil logiciel évalue les interfaces de type WUI. Il se base sur l'analyse du fichier log pour l'évaluation des pages web. Ce fichier contient un ensemble d'informations sur les événements interface-utilisateur. Au niveau des informations capturées pour l'analyse, WebRemUsine comporte l'adresse IP de la machine ainsi

que son nom en plus que les informations capturées par RemUsine. Il requiert l'installation du JDK pour procéder à la capture des informations. Une de ses limites réside dans le fait qu'il n'est utilisable qu'avec les navigateurs « *Microsoft Internet Explorer* » et « *Netscape Communicator* ». Comme résultat d'évaluation, WebRemUsine permet de renseigner sur les pages visitées ainsi que l'historique de la navigation. WebRemUsine correspond comme acronyme à « ***Web REM****ote* ***US****er* ***IN****terface* ***E****valuator* ».

### 3.2.9 MESIA (Trabelsi, 2006; Trabelsi et Ezzedine, 2012)

MESIA est un acronyme pour « ***M****ouchard* ***E****lectronique dédié à l'évaluation des* ***S****ystèmes* ***I****nteractifs à base d'****A****gents* ». Il s'agit d'un mouchard électronique permettant d'évaluer des systèmes interactifs. Il est spécifique pour un système donné ; il s'agit du SAI[7](ayant une interface de type WIMP). Ce mouchard est développé sous forme d'application logicielle. Son principal point fort est le fait d'évaluer des systèmes interactifs à base d'agents. MESIA est composé d'agents pour procéder à l'évaluation. Chaque agent mouchard est couplé à un agent système à évaluer. Ainsi MESIA est composé par le même nombre d'agents que le système évalué, Figure 3.6. Le processus d'évaluation est basé sur une comparaison entre deux séquences d'utilisation. La première est spécifiée par l'évaluation (assistée par le concepteur du SAI) en tant que modèle de référence. La seconde, quant à elle, est capturée depuis l'interaction entre l'utilisateur et le SAI. L'évaluation est ainsi une comparaison entre ces deux modèles suivant le principe énoncé par (Senach, 1990). Elle permet de renseigner sur les difficultés que l'utilisateur a connues lors de l'interaction et l'ensemble de tâches non accomplies par ce dernier. MESIA capture les événements de bas niveau (les événements physiques au sens de (Hilbert et Redmiles, 2000)). Par conséquent, l'évaluateur se trouve face à une énorme quantité d'informations à analyser et à interpréter. L'évaluation avec MESIA est élaborée lors de la phase de test (phase finale du cycle de développement des systèmes interactifs). MESIA permet de renseigner sur les problèmes essentiellement relatifs au critère d'utilité. MESIA capture et analyse les événements physiques d'une façon automatisée. Par contre, la phase de critique est élaborée manuellement par l'évaluateur. Son usage s'avère assez difficile pour une première utilisation du fait qu'il ne fournit aucun moyen d'interactivité avec l'évaluateur. De plus, il ne fournit pas de rapport d'évaluation. En sortie, ses auteurs se contentent d'un ensemble de statistiques portant sur l'interaction entre l'utilisateur et l'interface évaluée (celle du SAI).

7. Il s'agit d'un système utilisé dans les salles de contrôle du trafic du réseau de transport Valenciennois (Bus et Tramway). Il permet d'afficher différentes informations telle que le retard, la localisation des véhicules sur le réseau et ainsi que la communication avec les passagers et les conducteurs de ces véhicules. Globalement, ce système est chargé de gérer les perturbations sur le réseau. Son usage est destiné aux régulateurs du réseau de transport. Pour les lecteurs intéressés, ils peuvent consulter (Ezzedine *et al.*, 2008; Trabelsi *et al.*, 2004).

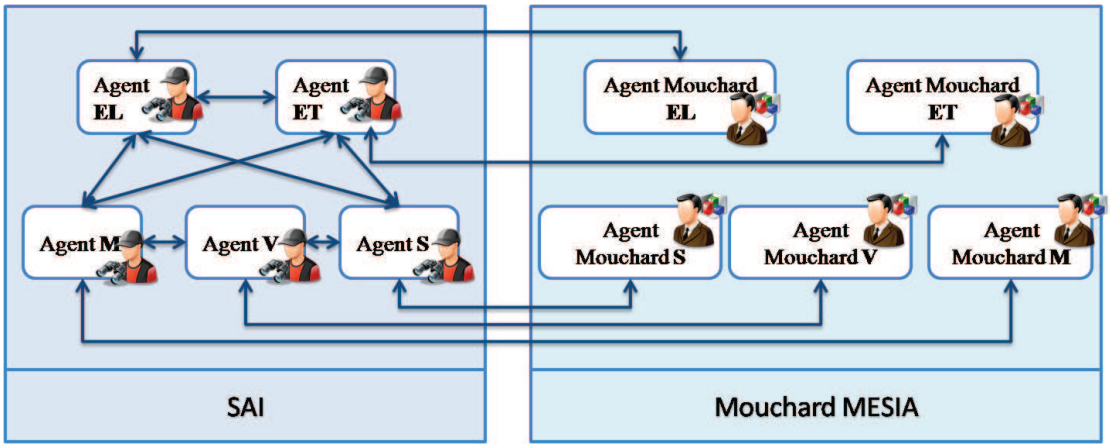



FIGURE 3.6: Principe de couplage entre les agents du mouchard et les agents du système évalué (Trabelsi, 2006)

### 3.2.10 MM WebRemUsine (Paterno *et al.*, 2007a)

MM WebRemUsine est un acronyme pour « ***M**ulti**M**odal **W**eb **REM**ote **US**er **IN**terface **E**valuator* ». Il correspond à un outil logiciel permettant de coupler entre plusieurs techniques d'évaluation dans le but d'inspecter l'utilisabilité dans les interfaces web : l'oculométrie (Beckers et Fuhr, 2010), l'analyse du fichier log et l'observation de la vidéo (via la webcam). Il est dédié pour l'évaluation des systèmes interactifs ayant une interface de type WUI. L'idée du couplage semble bien intéressante mais nous y retrouvons tout de même des difficultés pour l'évaluation. Tout d'abord, l'utilisation de cet outil d'évaluation s'avère coûteuse que ce soit au niveau de la mise en place ou encore au niveau de l'exploitation. En effet, MM WebRemUsine requiert du matériel tel est l'oculométre et la webcam. De plus, cet outil propose un nombre important d'informations à l'évaluation ; une tâche qui semble bien longue et pénible à traiter. De plus, (cf. Chapitre 1), l'oculométrie semble être délicate à exploiter (Beckers et Fuhr, 2010). Nous citons à titre d'exemple le cas d'une longue consultation qui semble assez fastidieux à exploiter. Certains pourront l'interpréter comme étant l'utilisateur se trouve perdu avec son interface ou encore désorienté. D'autres l'interpréteront différemment ; le contenu sur lequel s'effectue la fixation est intéressant. L'évaluation est automatisée lors de la capture ainsi que l'analyse. Par contre, la critique s'effectue manuellement.

### 3.2.11 Click Tale (C.E.Analytics, 2012)

Click Tale est un web service payant permettant de générer une sorte de carte de clic pour l'évaluation des interfaces utilisateurs de type WUI. Cette carte comporte les zones de l'interface sur lesquelles l'utilisateur a cliqué ainsi que la fréquence. Click Tale permet de capturer les actions physiques liées aux touches du clavier, le clic de la souris ainsi que

le mouvement du pointeur de la souris. En fin de capture d'événements, ce web service permet de générer deux sortes de rapports. Le premier est une heat map permettant de visualiser une synthèse des actions utilisateurs. Ce graphe permet d'indiquer sur la localisation du clic, le mouvement de la souris et la période de pointage sur une zone (où la couleur rouge correspond à une longue période et le bleue à une courte période), Figure 3.7. Le second rapport est un rapport qui comporte des données numériques et textuelles pour l'aide à l'évaluation des WUI. Il est généré par un outil dédié intitulé « *Form Analytics Tool* ». Click Tale permet même de rejouer les actions effectuées par l'utilisateur avec la souris. Cet outil permet à travers le second rapport d'identifier les aspects des sites engendrant une lenteur (ou encore un blocage) lors de l'interaction et les composants provoquant des erreurs d'utilisation. Le principal inconvénient de cet outil réside dans le fait qu'il ne permet pas la sauvegarde des données relatives à l'évaluation (sous forme de base de données ou autres). Ainsi il n'est pas possible de les consulter après l'évaluation.




FIGURE 3.7: Un exemple d'une heat map générée par ClikTale

### 3.2.12 Multidevice WebRemUSINE (Paterno *et al.*, 2007b)

Tout comme USINE et RemUsine, Multidevice WebRemUsine est outil destiné pour l'inspection de l'utilité dans les interfaces à travers l'analyse et l'exploitation des fichiers logs. Son usage est destiné pour les interfaces Mobile. L'évaluation des interfaces Mobile

soulèvent une difficulté qui réside essentiellement au niveau de la prise en considération du contexte d'utilisation lors de l'évaluation. En effet, la capture des informations relatives au contexte est une tâche difficile du fait qu'il change rapidement et facilement. Ainsi les auteurs de Multidevice WebRemUsine ont opté vers l'utilisation d'un outil intitulé « *Mobile Logger*[8] ». Cet outil permet de renseigner sur le contexte d'utilisation : localisation, degrés de luminosité, niveau de bruit, état de la batterie du dispositif mobile, puissance du signal du réseau, etc. De telles informations sont sauvegardées sous forme de fichiers XML que le Multidevice WebRemUsine exploite pour l'évaluation. L'évaluation via cet outil s'effectue une fois le système interactif est déjà conçu. L'évaluation est automatisée lors de la capture ainsi que l'analyse. Par contre, la critique s'effectue manuellement.

### 3.2.13 Gui Capture and Replay Tool (Jovic *et al.*, 2010; Jovic et Hauswirth, 2010)

Gui Capture and Replay Tool est un outil logiciel pour la capture des actions utilisateur avec des interfaces de type WIMP. Il consiste essentiellement à capturer et à rejouer les actions exécutées par l'utilisateur. Cet outil détecte les événements de bas niveaux tel que les touches du clavier ou encore les actions avec la souris (événements physiques au sens de (Hilbert et Redmiles, 2000)) sous forme de fichiers log. Ce fichier permet de rejouer les actions effectuées par l'utilisateur. Le visionnement de la séquence d'actions permet d'identifier les actions inutiles et les tâches que l'utilisateur n'a pas pu exécuter.

### 3.2.14 Click Heat (Labsmedia, 2012)

Click Heat est un web service open source permettant de tracer les zones de l'interface où les utilisateurs cliquent sur l'interface, Figure 3.8. Il permet d'évaluer les interfaces de type WUI par les utilisateurs finaux de ces interfaces lors de l'exécution des tâches ; autrement dit Click Heat propose la capture des clics utilisateur sur des interfaces web afin d'évaluer ces dernières. L'inconvénient avec ce web service réside dans le fait qu'il ne donne pas d'idée sur les utilisateurs. Ainsi l'évaluateur n'a même pas d'idée sur le profil utilisateur ainsi que le contexte d'utilisation ce qui rend l'exploitation et l'interprétation des données d'évaluation fastidieuse. L'utilisation de Click Heat pour l'évaluation requiert l'installation de scripts au niveau du navigateur côté utilisateur. Cet outil permet de renseigner sur le nombre de clic ainsi que leur localisation via des données numériques et des schémas graphiques. Il indique la fréquence de clic souris sur l'interface suivant une légende colorée allant du bleu au rouge où la couleur bleu correspond à un faible

8. www.willscullypower.wordpress.com

nombre de clics et la couleur rouge correspond au nombre le plus élevé de clics. L'interface proposée par Click Heat est parfaitement ergonomique au niveau du critère de guidage (au sens de (Bastien et Scapin, 1993)). Son usage est intuitif et facile pour un utilisateur (l'évaluateur) novice.

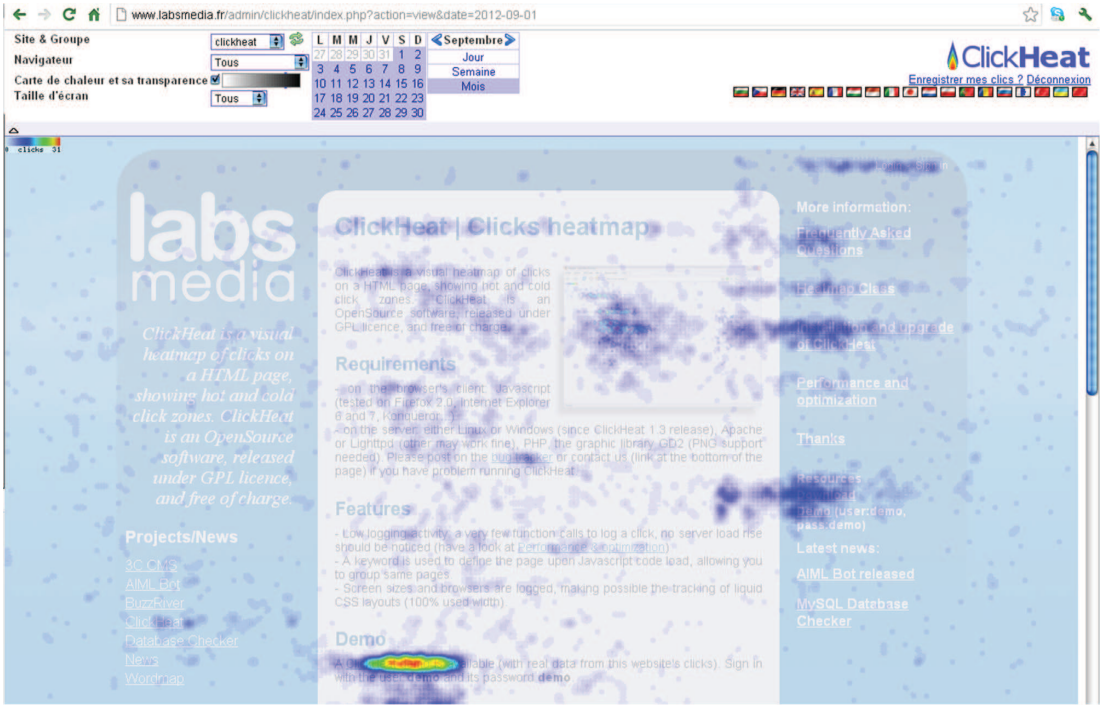



FIGURE 3.8: Exemple de heat map d'une page web générée par Click Heat (Labsmedia, 2012)

### 3.2.15 Morae (TechSmith, 2012)

Morae est un outil logiciel dédié pour l'évaluation des interfaces web. Il consiste à capturer toutes les données relatives à l'interaction entre l'utilisateur et l'interface : voix, vidéo, touche clavier, mouvement souris, clic souris, durée de consultation des pages web, etc.), Figure 3.9. Il est composé de trois modules. Ces derniers peuvent être installés sur trois machines différentes. Ces modules sont :

- Le module d'enregistrement (**Morae Recording Module**) : il se base sur des périphériques d'entrée/sortie pour détecter toutes les données citées ci-dessus tel est le cas d'un microphone, d'une webcam, etc. Notons que Morae supporte les navigateurs de type Internet Explorer et Firefox ;
- Le module d'observation (**Observer Module**) : il permet à l'évaluateur de noter ses observations de l'utilisateur interagissant avec l'interface à évaluer ;
- Le gestionnaire (**Manager Module**) : il s'agit du coeur du système d'évaluation. Ce module permet d'analyser, éditer et générer le résultat des données issues des deux premiers modules sous forme de rapport.

L'évaluation à travers cet outil s'avère agréable du fait que ce dernier est facile à manier et permet de dialoguer avec l'évaluateur via des messages interactifs. Néanmoins, le coût de l'évaluation s'avère relevé. En effet, d'une part l'évaluation requiert un nombre important de dispositifs comme cité au-dessus : webcam, microphone, etc. D'autre part sa licence est assez chère (elle coûte à peu près 1200€). Morae n'est utilisable que sous les systèmes d'exploitation Windows. A la fin de phase de capture, Morae propose un rapport des données détectées qui sont bien présentées et facilement exploitables par l'évaluateur.



FIGURE 3.9: Exemple d'interface de Morae

### 3.2.16 EISEval (Tran *et al.*, 2008)

EISEval est un environnement dédié pour l'évaluation des systèmes interactifs ayant une architecture à base d'agents. L'appellation EISEval correspond à un acronyme pour « ***E**nvironment for **I**nteractive **S**ystem **Eval**uation* ». EISEval s'articule autour de sept modules, Figure 3.10. Le premier module capture les données d'interactions dans une base de données « *MS Access 2007* ». Le second module associe les actions élémentaires en tâches que le module quatre générera les RdP associés. Le troisième module, quant à lui, consiste à analyser les données capturées. Le quatrième module permet de générer via l'outil « *Renew* » (Kummer *et al.*, 2003) les RdP issus des séquences d'interaction. Le cinquième module permet la confrontation entre deux RdPs (le modèle de référence et le modèle d'objet). Le module six fournit des critiques et des suggestions issues de l'évaluation, Figure 3.10. Enfin le septième module permet de configurer l'environnement pour l'évaluation des systèmes interactifs.

Tout comme MESIA, EISEval se base sur la capture des séquences d'interaction et en les comparant à un modèle de référence spécifié par le concepteur. La différence réside essentiellement en trois points. Le premier se trouve au niveau de la nature des

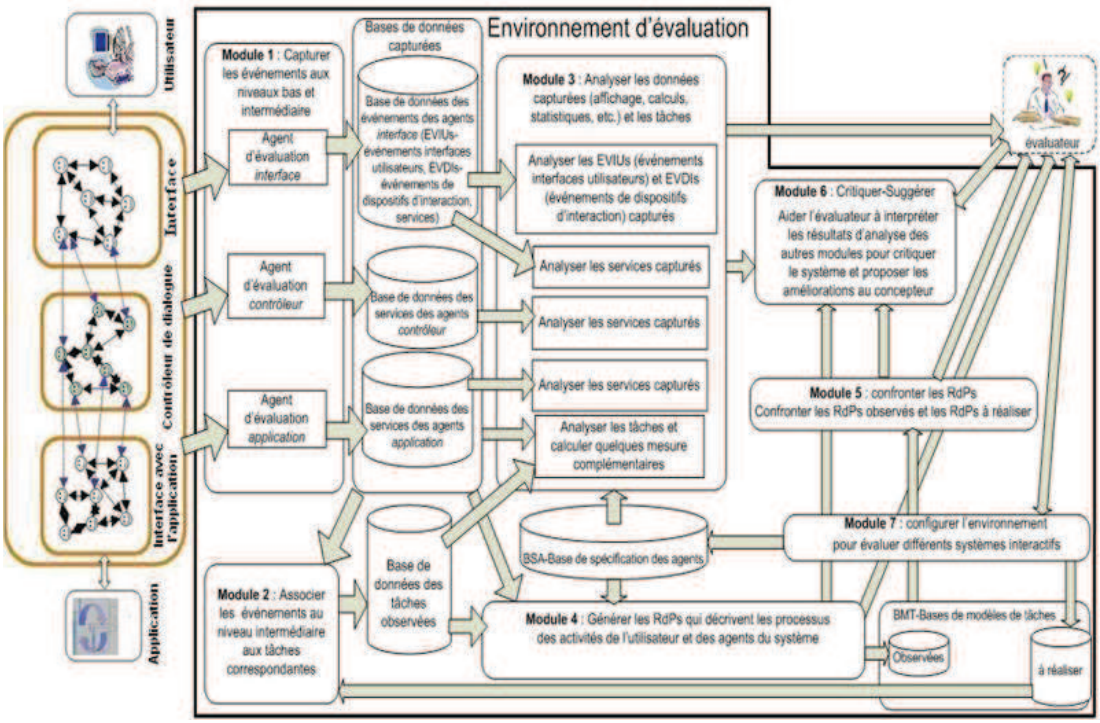



FIGURE 3.10: Architecture générale d'EISEval (Tran *et al.*, 2008)

événements capturés. EISEval capture les événements de type événements physiques, événements périphériques d'entrée et les événements interface utilisateur (au sens de (Hilbert et Redmiles, 2000)). Le second point est le fait qu'il soit générique pour l'évaluation des systèmes ayant une interface utilisateur de type WIMP. Le troisième point quant à lui concerne le fait que EISEval se base sur les réseaux de Pétri pour la modélisation et la comparaison des séquences d'action. Néanmoins, les résultats de comparaison sont deux RdP à comparer manuellement par l'évaluateur. De telle comparaison s'avère généralement pénible à effectuer vu la complexité parfois des RdP. Le processus d'évaluation requiert l'intervention de l'évaluateur, l'utilisateur et le concepteur (facultatif). L'évaluation est appliquée une fois le système interactif est conçu (ou encore dans l'état d'un prototype avancée). EISEval est un outil logiciel permettant d'inspecter l'utilité des systèmes interactifs. Il ne présente aucun rapport d'évaluation. L'évaluation est ainsi assistée par un ensemble de statistiques. Ces dernières sont tellement nombreuses qu'elles sont difficiles à exploiter. L'évaluation est automatisée durant les phases de capture et d'analyse. La critique quant à elle est effectuée manuellement par l'évaluateur. L'utilisation d'EISEval s'avère difficile pour un utilisateur novice du fait qu'il ne propose aucun moyen d'interactivité.

### 3.2.17 Feng-Gui (Shui, 2012)

Fen-Gui est un outil d'évaluation se basant sur l'étude du mouvement des yeux des utilisateurs interagissant avec une interface graphique. Le principal avantage de cet outil réside dans le fait qu'il ne fait pas appel à des vrais utilisateurs mais il se base sur des algorithmes issus de l'intelligence artificielle pour simuler l'action utilisateur. Ces algorithmes consistent à prédire ce que l'oeil de l'utilisateur va voire sur l'interface. L'idée forte derrière cet outil est le fait d'épargner l'évaluateur le travail de recherche et de sélection d'utilisateur pour son (ses) évaluation(s). A noter qu'ils existent certains outils dédiés à cet usage tel est le cas d'Ethnio [9].

Comme entrée, l'évaluateur propose à Feng-Gui des captures d'écran de l'interface à évaluer sous forme d'images que les algorithmes vont analyser et traiter. Comme avantage de cet outil, nous citons d'une part ses performances « *temps réel* ». En effet, quelques secondes suffisent pour analyser les images et simuler le regard de l'utilisateur, Figure 3.11. D'autre part, l'évaluation à travers Feng-Gui a l'avantage d'avoir un coût peu élevé. Néanmoins, nous n'avons pas une idée sur le degré de fiabilité de ces algorithmes du fait que nous ne disposons pas d'idée sur les algorithmes utilisés. En effet, l'approche adoptée est critiquable du fait que les données sont traitées suivant une approche descendante. De plus, cet outil n'est pas assez interactif d'où une difficulté d'utilisation. Feng-Gui vise à inspecter essentiellement l'utilisabilité dans les interfaces utilisateur. Le processus d'évaluation à travers cet outil est automatisé lors de la capture. Par contre, l'analyse et la critique s'effectuent manuellement.

9. Ethnio n'est pas un outil d'évaluation des interfaces utilisateurs. Il s'agit d'un simple moyen pour recruter les utilisateurs afin de procéder à des tests utilisateurs et de donner leur feedback. Ethnio permet de résoudre les problèmes de choix d'utilisateurs. La recherche d'utilisateur peut s'effectuer suivant différents critères que cet outil propose : âge, sexe, profil, etc. (Peters, 2012).

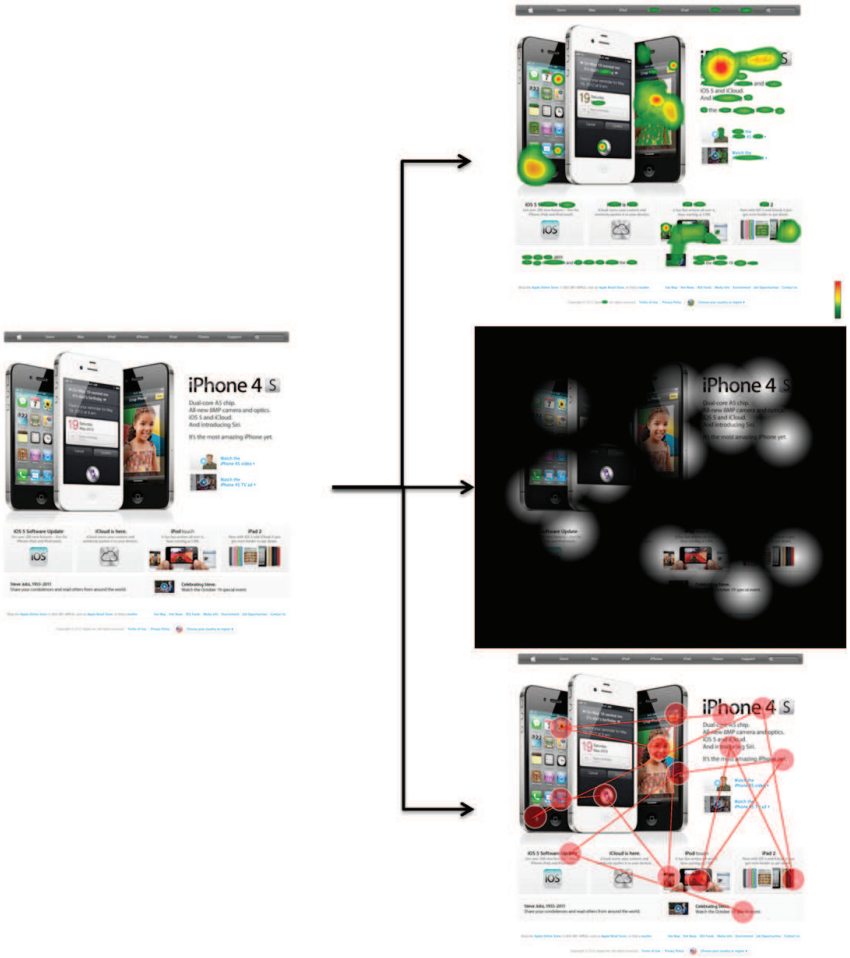



FIGURE 3.11: Exemple d'image analysée par Feng-Gui

### 3.2.18 Crazy Egg (CrazyEgg, 2012)

Crazy Egg est web service basé sur la détection et le traçage du clic sur la souris pour l'évaluation des interfaces graphiques de type WUI. Il permet de renseigner l'évaluateur sur tous les détails portant sur le clic souris. Afin de capturer de telles interactions, du code java script doit être ajouté au niveau du code source des pages web évaluées. En fin de processus de capture, Crazy Egg propose un rapport bien détaillé. De plus, la carte de clic est bien présentée et facile à exploiter pour l'évaluation, Figure 3.12. Crazy Egg propose même de partager le rapport d'évaluation sous forme de lien URL. Les données capturées peuvent être travaillées d'avantages du fait que l'évaluateur peut les sauvegarder sous format de fichier Excel ou dans des bases de données. Les auteurs de Crazy Egg proposent un outil permettant de mieux présenter et exploiter les données d'évaluation. Il s'agit de « *Confetti* ». L'auteur de cet outil (Neil Patel) affirme même sur leur blog grâce à Confetti ils se sont aperçus que les utilisateurs du navigateur Firefox agissent différemment que ceux utilisant le navigateur Internet Explorer. Ce que nous

reprochons à cet outil le fait que l'évaluateur ne peut procéder à l'évaluation des sites web que page par page. De plus, il ne permet pas d'interroger les utilisateurs pour justifier leur action ou encore pour dégager leur appréciation.

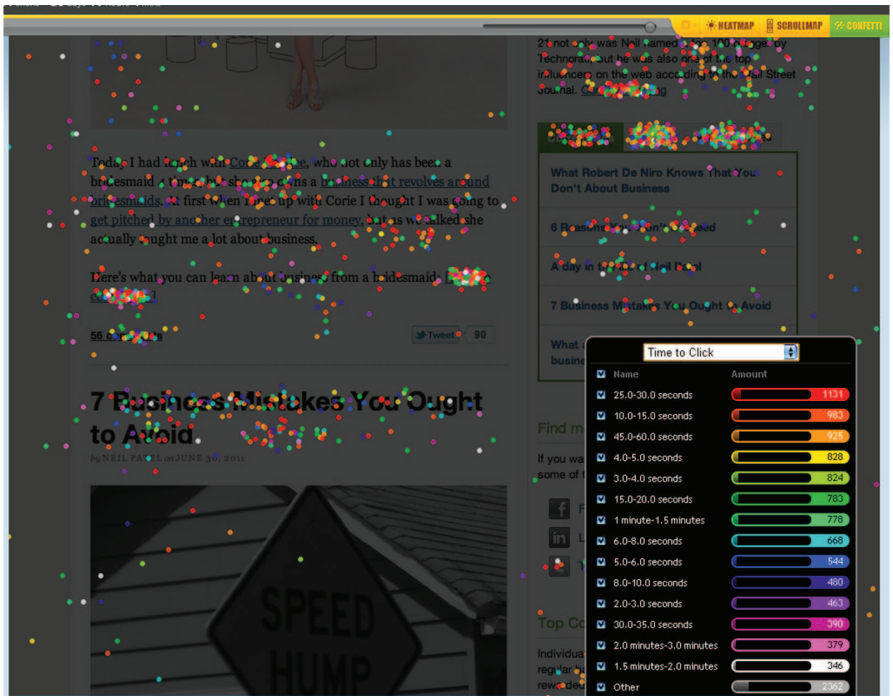



FIGURE 3.12: Exemple d'une heat map issue de l'évaluation d'une page web avec Crazy Egg

### 3.2.19 Clixpy (Clixpy, 2007)

Clixpy est un web service permettant la capture des actions physiques de l'utilisateur avec une interface web. Il permet de capturer toutes les actions sauf le contenu des champs de texte de type mot de passe (mouvement du curseur, clic souris, entrée dans les champs de texte, défilement souris, etc.). Afin de capturer ces informations, un code JavaScript doit être intégré au niveau du code HTML de l'interface à évaluer. Clixpy est un outil facile à utiliser ou toutes les actions et les opérations sont clairement indiquées à l'évaluateur. Néanmoins, ce que nous reprochons à cet outil est la qualité du rapport d'évaluation fourni à l'évaluateur. En effet, il est bien difficile à exploiter pour l'évaluation des WUI. De plus, l'évaluateur peut ré-visionner la session d'interaction capturée. La capture à travers cet outil s'effectue manuellement. Par contre, les phases d'analyse et de critique sont laissées aux soins de l'évaluateur.

### 3.2.20 Open Hallway (Hallway, 2012)

Open Hallaway est web service dédié pour l'aide à l'évaluation des sites web. Ce web service consiste à élaborer des scénarios d'évaluation. Il permet par la suite de capturer des sessions d'interaction avec l'utilisateur. Parmi les avantages de cet outil, nous citons le fait que l'évaluateur n'a pas à ajouter des scripts ou encore du code au niveau du code HTML des pages à évaluer. De plus, aucun téléchargement d'outils supplémentaires ou bien de plugin est requis pour l'évaluation (en effet, à part le JDK installé sur l'ordinateur, Open Hallaway ne requiert pas d'autres installations). Open Hallway consiste à capturer des données vidéo et audio lors de l'interaction avec l'utilisateur afin de les exploiter pour l'évaluation des sites web. Ces données sont sauvegardées sur un serveur et peuvent être téléchargées par l'évaluateur sous forme de fichier de format mp4. L'inconvénient de telle approche réside dans le fait qu'elle ne supporte pas les interfaces multi-écrans et que les vidéos capturées ne doivent pas dépasser les dix minutes. Le processus d'évaluation proposé est assez facile à mettre en place. En effet, en entrée, Open Hallway requiert le lien URL de la page à évaluer et ensuite il génère un lien URL à transférer à (aux) utilisateur(s). Puis, Open Hallway génère l'enregistrement de la session d'interaction pour l'exploiter à l'évaluation de l'interface utilisateur. Ainsi, cet outil n'est automatique que lors de la capture. L'analyse et la critique sont effectuées manuellement par l'évaluateur.

### 3.2.21 Usabilla (Usabilla, 2012)

Usabilla est un web service pour évaluer les interfaces web. Il permet de collecter le feedback des utilisateurs et de détecter les clics de l'utilisateur lors de l'interaction. Son principal point fort réside au niveau de la possibilité d'envoyer des messages tel que « *Cliquer sur l'endroit que vous préférez le plus* » pour détecter l'appréciation de l'utilisateur vis-à-vis de la page évaluée. Usabilla propose à l'évaluateur un ensemble prédéfini de questions qu'il peut changer en ajoutant ou bien modifiant quelques questions. De plus, Usabilla permet à l'utilisateur de partager leur avis. Ainsi ce web service se base sur des données qualitatives et quantitatives pour l'inspection de l'utilisabilité des sites web, Figure 3.13. La capture des événements utilisateurs (événements physiques au sens de (Hilbert et Redmiles, 2000)) s'effectuent grâce à l'intégration de code au niveau de l'interface web à évaluer. A partir des clics souris détectés, Usabilla génère une carte de clic que l'évaluateur peut la sauvegarder sous différents formats (PDF, CSV, TXT, XML, etc.). Son principal point fort réside essentiellement dans le fait que l'évaluateur peut demander des questions via cet outil à l'utilisateur lors de l'interaction et qu'il est facile à exploiter pour l'évaluation. De plus, il est utilisable à n'importe quelle phase du

cycle de conception des sites web. En entrée, il requiert l'URL de la page web ou encore une simple image de ce site. Ainsi, l'évaluateur peut évaluer un croquis d'un site web avec cet outil.

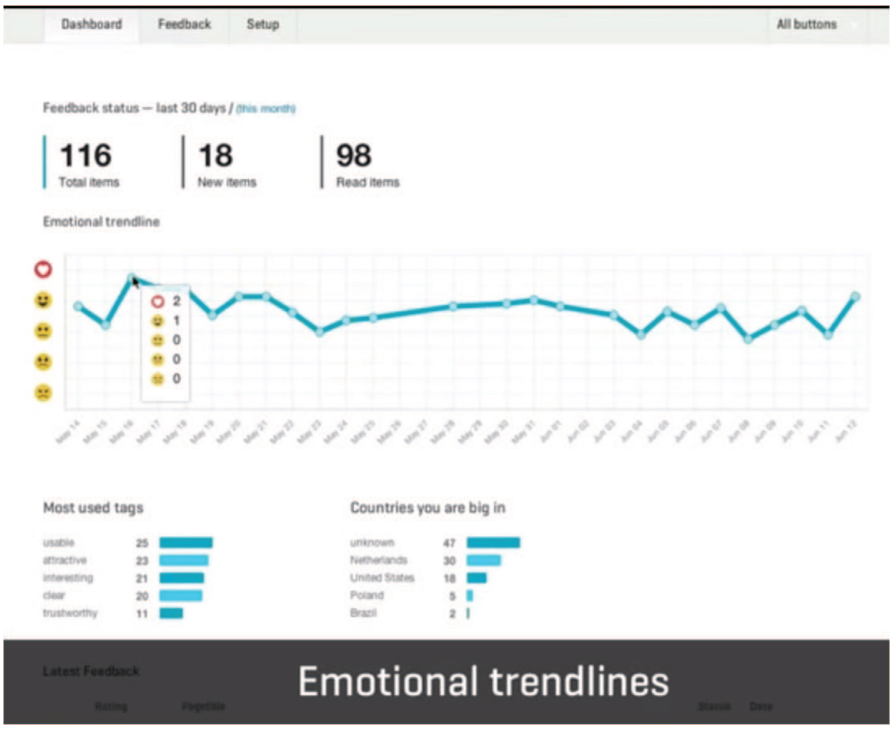



FIGURE 3.13: Un exemple de heat map généré par Usabilla los de l'évaluation d'un site web.

### 3.2.22 Userfly (Userfly, 2009)

Userfly est un web service dédié à l'inspection de l'utilisabilité des sites web via des tests à distance. Il permet de capturer des données portant sur l'utilisation de la souris par l'utilisateur : le mouvement du curseur et le clic. Il permet de rejouer l'interaction effectuée par l'utilisateur. Afin de capturer de telles données, Userfly requiert l'ajout de script Java Script au niveau de la source de l'interface web. En sortie, Userfly propose un rapport détaillée qui comporte un ensemble important de données numériques : pages visitées, temps passé pour chaque page, texte entré, clic de la souris, défilement avec la souris, etc. L'utilisation de cet outil s'avère facile. Néanmoins, les données capturées sont sauvegardées pour une période de 90 jours seulement sur le serveur et les auteurs de cet outil ne proposent pas de moyens de sauvegarde de ces données. De plus, il n'est pas possible d'interagir entre l'évaluateur et l'utilisateur pour répondre à certaines questions soulevées par l'interaction entre l'utilisateur et l'interface. Userfly propose une évaluation assez similaire à celle proposée par Clixpy. La différence réside essentiellement à l'interactivité d'Userfly. En effet, il propose des interfaces plus interactives. Ainsi son utilisation est plus facile.

### 3.2.23 EVALYZER (Insydelabs, 2013)

EVALYZER est un outil web permettant d'élaborer des tests utilisateur à distance des sites web afin d'évaluer ces derniers. Il assure la collecte des données relatives au comportement des internautes d'une façon exhaustive. L'évaluation à travers cet outil est structurée autour de sept étapes :

1. Définition des tâches à exécuter par l'utilisateur ;
2. Création d'un questionnaire dédié pour la collecte d'information utilisateur ;
3. Paramétrage du test utilisateur (nature des données capturées et l'ordre des tâches à détecter) ;
4. Définition de la liste des actions à exécuter par l'utilisateur ;
5. Invitation des participants aux sessions de tests ;
6. Visualisation des données capturées et,
7. Identification des problèmes de conception du site web évalué par l'évaluateur.

Afin d'assister l'évaluateur, EVALYZER lui propose les données capturées sous forme graphique assez facile à exploiter. Les données proposées sont essentiellement le taux de réussite, d'échec et d'abandon des actions exécutées par l'utilisateur. Il propose aussi les vidéos enregistrées lors de l'interaction. EVALYZER considère les actions de haut niveau ce qui évite à l'évaluateur un flux important d'informations qui requièrent dans la majorité des cas un temps et un effort importants de la part de l'évaluateur pour leur exploitation. L'exploitation d'EVALYZER pour l'évaluation est possible pour un site web fini et prêt à l'utilisation. La capture et l'analyse des données sont établies automatiquement par l'outil d'évaluation. La phase de critique quant à elle est élaborée manuellement par l'évaluateur.

### 3.2.24 Loop11 (Loop11, 2012)

Loop11 est un service web dédié à l'évaluation des interfaces web. L'évaluation avec ce service web consiste à effectuer des tests d'utilisabilité à distance. Cet outil permet de supporter l'évaluation via la présentation de données métriques portant essentiellement sur le taux d'achèvement des tâches, le temps d'exécution des tâches par les utilisateurs, etc. Un des avantages de cet outil est le fait qu'il est facile à utiliser par un utilisateur novice ou encore un non expert de l'évaluation des sites web. Ainsi son utilisation peut s'effectuer par les concepteurs des sites web afin de tester leurs sites web. Les informations sont bien présentées et son interface est bien agréable à utiliser. De plus, Loop11 permet de créer des questionnaires afin de recueillir les attentes et les impressions des utilisateurs. Les résultats d'évaluation sont bien présentées et faciles à exploiter, Figure 3.14. Cet outil

propose d'exporter ces résultats sous forme de rapport en CSV, XML et PDF. Loop11 ne permet d'évaluer que les sites déjà conçus. Parmi les avantages de cet outil, nous pouvons citer :




FIGURE 3.14: Exemple de résultat de capture de données avec Loop11

– facilité de l'élaboration des tests utilisateurs ;
– pas de codage ni de logiciel à installer ;
– des résultats en temps réel (il ne requiert pas beaucoup de temps pour être analysés) et,
– générique pour n'importe quelle interface WUI.

Tout comme la majorité des outils présentés dans ce chapitre, Loop11 permet une capture et une analyse totalement automatisées. Mais par contre, la phase de critique s'effectue manuellement par l'évaluateur.

### 3.2.25 Silverback (Ltd, 2012)

Silverback est un outil logiciel utilisable que sous les systèmes d'exploitation « *Mac OS* ». Il permet de capturer, analyser, éditer et rejouer les sessions de test d'un site web. Silverback est un outil facilement utilisable pour les évaluateurs novices. Son principe consiste à associer entre les captures d'écran et des enregistrements audio et vidéo. Il permet

de renseigner sur la façon avec laquelle l'utilisateur interagit avec l'interface. Silverback ne permet pas de donner d'avantage d'informations concernant la session d'évaluation. Afin de continuer l'évaluation, l'évaluateur doit visionner l'enregistrement tout en se focalisant sur les séquences d'interaction et l'enregistrement vidéo de l'utilisateur, Figure 3.15. L'évaluation s'avère ainsi assez délicate et relève de l'interprétation de l'évaluateur que d'autre chose. Silverback ne permet pas de donner des mesures quantitatives portant sur la session d'interaction enregistrée.

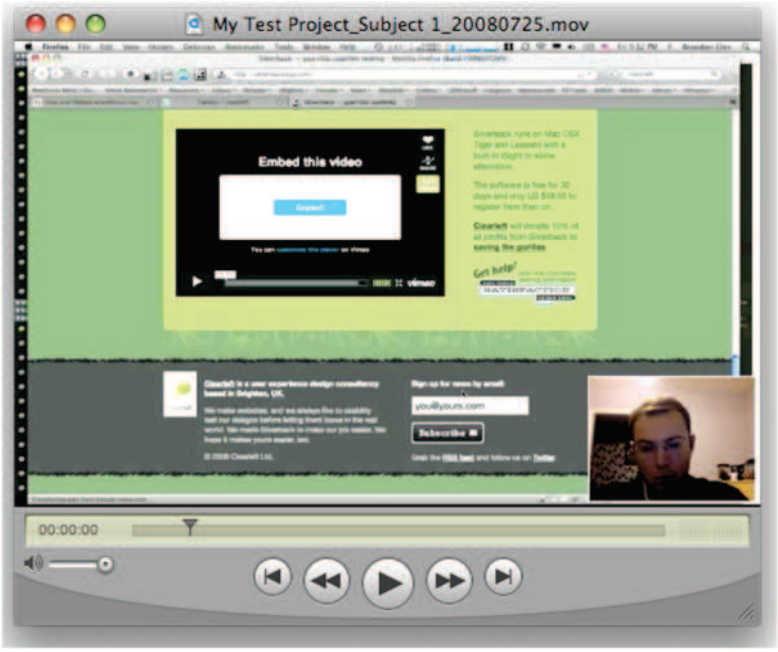



FIGURE 3.15: Silverback (Photo extraite de www.Gigaom.com).

## 3.3 Synthèse et discussion

Il existe de nombreux outils d'évaluation des interfaces utilisateurs exploitant l'interaction entre l'utilisateur et l'interface à évaluer (cf. Annexe D : Tableau D.1, Tableau D.2, Tableau D.3 et Tableau D.4. Ces outils sont divers et peuvent varier suivant plusieurs critères. De point de vue nature de l'interface évaluée, la majorité des outils proposés ne s'intéressent qu'à l'évaluation des interfaces WUI (15 outils parmi 24). Ceci s'explique d'une part par l'évolution croissante du monde du web. De plus, il parait bien plus facile d'évaluer une interface WUI qu'une interface WIMP ou encore Mobile du fait que l'évaluateur peut facilement accéder à son code source (codé en HTML) et procéder à l'évaluation. De plus, généralement le nombre du public cible des interfaces web dépassent de loin celui d'une interface WIMP. Néanmoins, les outils pouvant évaluer ou encore assister à l'évaluation des interfaces Mobile sont encore limités (point de vue nombre). Ceci est en désaccord avec l'intérêt qu'accordent les communautés du

génie logiciel, ergonomie logicielle et de la programmation. En effet, de telles interfaces sont de plus en plus répandues suite à la « *démocratisation* » de la technologie mobile et sa propagation dans tous les secteurs. Parmi les outils présentés, seul Multi Device RemUsine permet l'évaluation des interfaces Mobile. Au niveau de la nature de l'outil d'évaluation, les outils sont répartis en deux groupes. Ces deux groupes sont le web service et l'outil logiciel (parmi 24 outils : 12 outils sont implémentés en web service et les 12 autres sont implémentés sous forme d'application logicielle). Cette répartition s'explique par le fait que les solutions à base d'architecture orientée service (Perrey et Lycett, 2003) (SOA) est de plus en plus répandues an niveau de la communauté logicielle du fait qu'elle garantit une meilleure interopérabilité et résout les problèmes de compatibilité que les outils peuvent connaitre (Newcomer et Lomow, 2004). A titre d'exemple, nous citons l'exemple de Silverback. Cet outil, comme cité ci-dessus, n'est utilisable que sous le système d'exploitation « *Mac OS* ». Quant à l'ensemble des intervenants dans le processus d'évaluation, la majorité d'outils requièrent la présence de l'utilisateur (ce qui est évident du fait qu'ils se basent sur l'interaction entre l'utilisateur et l'interface) et l'évaluateur. Au niveau du principe utilisé lors de l'évaluation, il est intéressent de remarquer la diversité des principes utilisés. Une première collection d'outils se base sur des comparaisons entre différentes séquences d'utilisation suivant le principe d'évaluation énoncé par Senach (Senach, 1990). Cette collection regroupe six outils. La seconde se base sur l'observation des sessions d'interaction pour dégager les éventuels problèmes d'utilisabilité. Cette collection comporte dix outils. La troisième collection se base sur la capture et l'étude des mouvements de la souris. Il s'agit de « *Mouse Tracking Tool* ». Les outils issus de cette collection se basent sur la dissociation entre direction du regard et localisation du pointeur de la souris. Le pointeur permet ainsi d'avoir une idée claire sur les points d'intérêts de l'utilisateur au niveau de l'interface et par suite de dégager les informations ou encore les zones auxquelles l'utilisateur ne leur prêtent pas d'intérêt. Cette collection regroupe sept outils. Au niveau de l'exploitation des données détectées, la première collection semble donner plus de facilité au niveau du traitement des informations. En effet, nous remarquons que ses résultats sont généralement générés d'une façon automatique et l'évaluation relève de l'objectivité. Par contre ; la deuxième et la troisième collection relève plus des interprétations de l'évaluation et par conséquent de la subjectivité. Nous constatons aussi que la majorité des outils d'évaluation se focalisent sur des événements de bas niveau pour l'évaluation des systèmes interactifs. Ces événements sont généralement des événements physiques (au sens de (Hilbert et Redmiles, 2000)). Ceci s'explique par le fait qu'il est facile de capturer les événements de bas niveau. Leur répartition suivant des tâches, des domaines et des niveaux abstraits de l'interaction requiert l'utilisation des outils spécifiques à la modélisation des tâches utilisateur tel que le cas avec CTTE. Il est aussi intéressent de constater que la totalité des outils d'évaluation présentés ne sont applicables que lors de la dernière phase du

cycle de développement des systèmes interactifs : la phase de test. Ceci peut expliquer le fait que certains travaux (Grislin et Kolski, 1996) confirment que la phase d'évaluation est négligée dans la majorité des cas lors de l'élaboration de systèmes interactifs. En effet, comme elle n'est pas prise en considération depuis les premières phases de conception, elle peut être abandonnée suite au problème de délai et de budget. De plus, son implémentation pourra être plus coûteuse si elle n'est prise en considération qu'à la fin du cycle. En effet, Nielsen affirme qu'il revient 100 fois moins cher de procéder à une évaluation précoce qu'une évaluation lors de la dernière phase de conception (Nielsen, 1993). Nous pouvons aussi nous focaliser sur le degré d'automatisation des outils d'évaluation. Tout comme les outils d'évaluation présentés lors du second chapitre, la majorité des outils présentés dans ce chapitre offre une capture et une analyse automatisées par contre la critique reste au bon soin de l'évaluateur. Ceci peut atténuer la fiabilité de l'évaluation. En effet, elle dépendra ainsi de l'évaluation, son jugement ainsi que son expérience. Certaines recherches ont confirmé la variation des résultats de recherche suivant le niveau d'expertise et l'expérience de l'évaluateur. Cette variation s'avère dans la majorité des cas très importante (Jacobsen *et al.*, 1998). Comme fonctionnalités des outils d'évaluation présentées, la grande majorité d'entre elles propose des données qui peuvent être exploitées pour l'évaluation. En effet, ces outils proposent un ensemble de statistiques, des graphiques (tel que la heat map), des notes ou encore des vidéos de sessions d'interaction que l'évaluateur doit interpréter pour dégager les problèmes d'utilité ou encore d'utilisabilité dans l'interface évaluée. Les données d'évaluation peuvent être très hétérogènes et peuvent provenir de différentes sources : test à distance, des rapports d'évaluation, des tests en laboratoire, etc. Ainsi une difficulté d'exploitation en résulte. Aucun outil de ceux présentés ne permet de générer directement une critique automatique de l'interface. Nous constatons dans ce chapitre une grande variété d'outils d'évaluation. Néanmoins ces outils se contentent de capturer et enregistrer les actions utilisateurs sans proposer de moyen ou encore de méthodologie pour les exploiter. Les travaux dans ce contexte sont assez limités. De ce fait, beaucoup de recherche ont récemment porté sur l'exploitation des données issues des méthodes de recherches. Ces travaux se sont focalisés sur la façon avec laquelle les données seront utilisées pour l'évaluation. Par exemple, Wixon s'est focalisé à assurer le lien entre ces données et les facteurs contextuels (Wixon, 2003). D'autres travaux ont porté sur les éléments à inclure lors d'un rapport d'évaluation afin de mieux répondre aux attentes et besoins de l'évaluateur (Theofanos et Quesenbery, 2005). Certains travaux consistent à formaliser les problèmes d'interaction et à proposer des formats pour les décrire tel est le cas des travaux de Cockton et Lavery (Cockton *et al.*, 2004; Lavery *et al.*, 1997). Howarth a démontré l'intérêt de telle formalisation et transformation des données de l'évaluation en procédant à l'évaluation d'un système avec deux groupes d'évaluateurs novices. Les premiers ont opté pour l'observation d'une séquence vidéo de session d'interaction. Le

second groupe a procédé à l'implémentation d'un outil permettant de transformer et de formaliser les données capturées (qu'il qualifie dans son article par données brutes). La comparaison de résultats d'évaluation des deux groupes a montré que celle du second groupe est plus fiable en termes de problèmes d'interfaces détectés.

## 3.4 Conclusion

Ce chapitre a été dédié à une revue d'outils d'évaluation de la partie interactive des systèmes interactifs. Ces outils ont la caractéristique de se baser sur l'interaction entre l'utilisateur et le système interactif. Le présent chapitre est une continuité du second chapitre qui a présenté les outils se basant sur les règles ergonomiques pour l'évaluation des systèmes interactifs. Les outils présentés, comme mentionnés en section 3.3, se diversifient suivant de nombreux critères : critères inspectés, méthodes de recueil des données, fonctionnalités supportées, degré d'automatisation, etc. Dans la partie suivante, nous nous proposons d'introduire une nouvelle contribution pour l'évaluation des systèmes interactifs.

# Deuxième partie

# Contribution

Chapitre 4

# Contribution 1 : Le Framework RITA pour l'évaluation des interfaces utilisateurs

> « *Simplicity : An interface should be simple (not simplistic), easy to learn, and easy to use. It must also provide access to all functionality of an application. Maximizing functionality and maintaining simplicity work against each other in the interface. An effective design balances these objectives.* »
>
> (Microsoft, 2011)

## Introduction

Au cours des deux chapitres précédents, nous avons vu qu'ils existent de nombreux outils pour l'évaluation des systèmes interactifs. Ces outils ont été présentés suivant deux catégories. La première s'intéresse à l'évaluation de la présentation statique de la partie interactive d'un système interactif. La seconde implique l'utilisateur lors du processus d'évaluation ; les outils relatifs à cette catégorie se basent sur la capture, l'analyse et l'étude des actions utilisateur pour l'évaluation.

Dans ce chapitre nous allons introduire un framework qui évalue les systèmes interactifs suivant trois techniques d'évaluation tout en se focalisant sur la présentation statique de l'interface et de l'interaction utilisateur-système. Ce chapitre est structuré suivant quatre sections. La première section consiste à présenter globalement le framework d'évaluation proposé. Ensuite, lors de la seconde section, l'architecture fonctionnelle du framework

est présentée. Par la suite, la troisième section comporte la description du processus d'évaluation des systèmes interactifs adopté avec le framework qui est présenté dans ce chapitre. Enfin, une discussion est établie lors de la quatrième section. Elle consiste essentiellement à situer le framework présenté par rapport aux travaux existants et à identifier ses limites. Le framework présenté dans ce chapitre est baptisé *RITA* pour Use**R**-**I**nterface Evalua**T**ion fr**A**mework.

## 4.1 Présentation générale du framework

### 4.1.1 Motivation

Nous avons constaté lors de la première partie de ce mémoire que la majorité des outils existants pour l'évaluation des systèmes interactifs se base essentiellement sur une seule technique d'évaluation. Chaque technique a sa spécificité pour l'évaluation et considère un nombre d'aspect différents des autres techniques (Nielsen, 1993). Afin de procéder à l'évaluation, il serait plus intéressant et avantageux de se baser sur plus d'une technique d'évaluation (Nielsen, 1993; Charfi *et al.*, 2009; Farenc *et al.*, 2001). De tel concept constitue la motivation majeure pour la mise en place d'un framework se basant sur différentes techniques d'évaluation : le mouchard électronique, le questionnaire dédié à l'évaluation des *IHM* et l'inspection de la qualité ergonomique d'une interface. De plus, nous constatons que généralement chaque outil d'évaluation ne permet d'évaluer qu'un seul type d'interface (WIMP, WUI ou Mobile). De ce fait, nous nous estimons qu'il serait intéressent de concevoir un système d'évaluation multi-techniques destiné pour un usage multi-interfaces. La présente contribution n'est qu'un effort pour étendre la gamme de fonctionnalités des outils existants pour l'évaluation des *IHM*.

### 4.1.2 Concept

Cette contribution vise à promouvoir un framework (kit logiciel) d'évaluation des systèmes interactifs. Ce framework est :

- **Générique.** Il permet d'évaluer les interfaces de différents systèmes interactifs ;
- **Configurable.** Il est structuré suivant une architecture modulaire ce qui permet sa configuration pour l'évaluation de différents systèmes interactifs ;
- **Flexible.** Ce framework se base sur les règles ergonomiques pour l'évaluation de la qualité ergonomique des systèmes interactifs. Ces règles ne sont pas codées en dur dans le moteur d'évaluateur ce qui permet une flexibilité au niveau de la mise à jour de la base des règles à prendre en considération lors de l'évaluation ;

- **Multicritères.** Le framework proposé se base sur l'évaluation de la présentation statique des interfaces utilisateur, l'interaction entre l'opérateur Humain et le système et ainsi que la satisfaction de l'utilisateur vis-à-vis du système évalué.
- **Multi-techniques.** Le framework proposé se base sur trois techniques différentes d'évaluation :
  - Le mouchard électronique. Le mouchard permet de recueillir en situation réelle des actions utilisateur dans le but de les analyser pour identifier les problèmes d'utilisation dans l'interface.
  - L'inspection de la qualité ergonomique dans l'interface. L'inspection consiste à valider la conformité de l'interface par rapport à un ensemble de règles ergonomiques pour détecter les incohérences sur l'interface afin de les corriger et par suite d'obtenir des interfaces utilisateurs de bonne qualité ergonomique. Les règles considérées lors de l'évaluation sont spécifiées préalablement par l'évaluateur.
  - Le questionnaire. Il permet de demander à l'utilisateur son avis sur l'interface évaluée et/ou d'inspecter les aspects dont il a été impossible de les inspecter via le mouchard et l'inspection de la qualité ergonomique (tel est l'exemple de la clarté des messages d'erreurs).

## 4.2 Architecture fonctionnelle du framework

Le framework proposé s'articule autour de quatre modules : un module gestionnaire de règles ergonomiques, un module de capture des *IHM*, un moteur d'évaluation et un outil de génération de rapport, Figure 4.1. Le framework d'évaluation est structuré suivant une architecture modulaire dans le but de garantir une indépendance entre les différents éléments du framework et par la suite de permettre, sans configuration, l'évaluation des SI.

### 4.2.1 Module 1 : Gestionnaire de règles ergonomiques

Ce module permet la gestion des règles ergonomiques pour l'évaluation des systèmes interactifs. Une règle est définie par, Figure 4.2 :

- **Identifiant :** permet de faciliter l'opération d'exploitation de la règle ainsi que sa manipulation (modification, consultation, etc.). L'identifiant est géré automatiquement par le système ;
- **Libellé :** permet d'attribuer une nomenclature à la règle ;
- **Description :** consiste à décrire la règle et d'énoncer son principe pour la conception ou bien l'évaluation des *IHM* ;

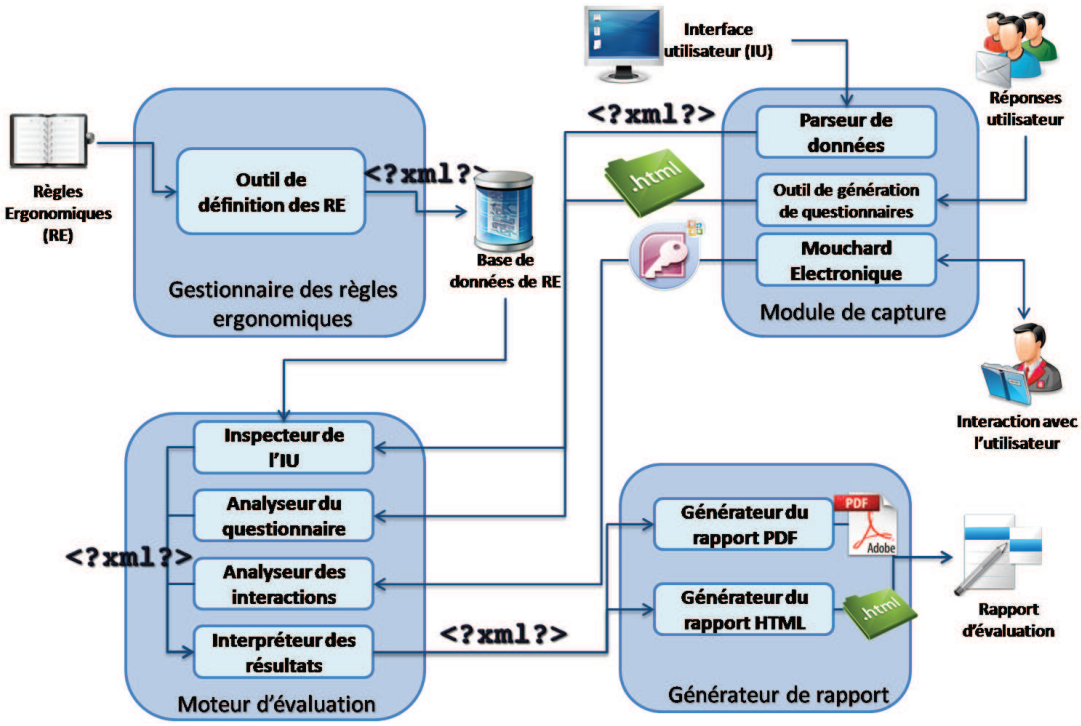



FIGURE 4.1: Architecture du framework RITA

- **Erreur :** permet de renseigner sur l'erreur engendrée lors du non respect de la règle dans l'interface utilisateur ;
- **Recommandation :** permet de fournir des recommandations à l'évaluateur (ou concepteur) pour remédier au non respect de la règle dans l'interface utilisateur ;
- **Technique d'évaluation :** renseigne sur la technique spécifiée pour inspecter la conformité de l'interface par rapport à cette règle ;
- **Critère (sous critère) ergonomique :** que la règle permet d'inspecter lors de la validation de la conformité de l'interface par rapport à ce critère (sous critère). Nous adoptons ici les critères et les sous critères de Bastien et Scapin (Scapin et Bastien, 1997).
- **Contexte :** consiste à spécifier le contexte d'utilisation à prendre en considération lors de l'évaluation de l'*IHM* ;
- **Exemple positif (négatif) :** consiste à illustrer un exemple d'une interface où la règle est respectée (non respectée) ;
- **Corpus :** d'où provient la règle ergonomique et,
- **Référence :** la référence bibliographique où est mentionnée la règle.

L'évaluateur peut à travers ce module spécifier les règles sous forme de fichiers XML. Ainsi, la base de règles ergonomiques est un ensemble de fichiers XML que l'évaluateur peut modifier. Les règles ergonomiques sont spécifiées en associant des valeurs de l'attribut d'un (des) composant(s) de l'interface à des valeurs numériques. Cette association est faite grâce à des opérateurs mathématiques, Figure 4.3. Ces opérateurs sont :

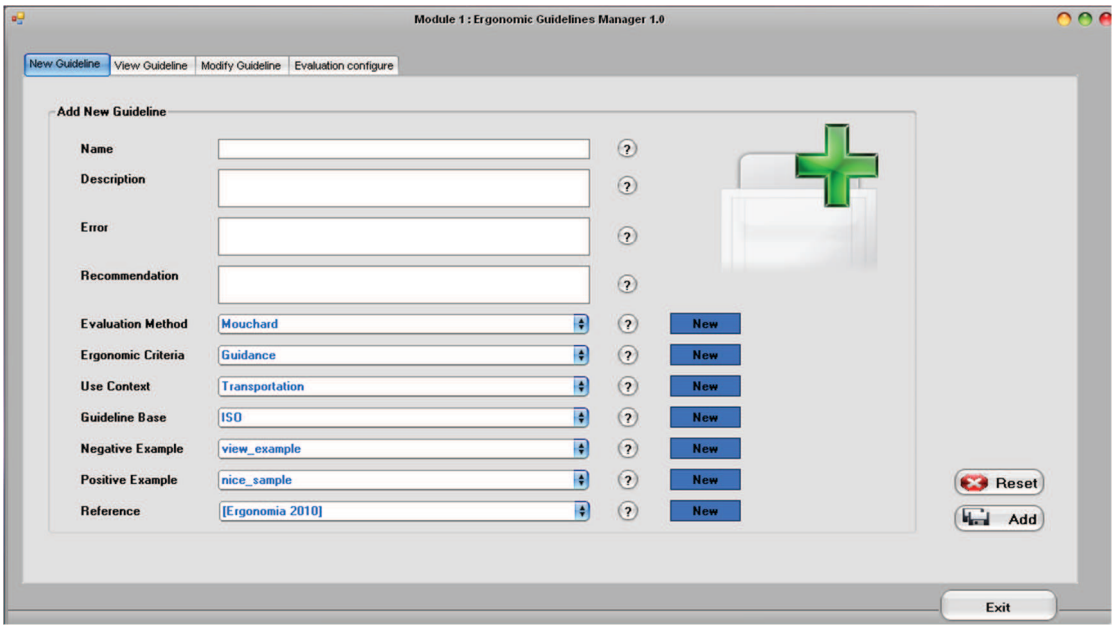



FIGURE 4.2: Gestionnaire des règles ergonomiques : interface permettant l'ajout de nouvelles règles

- **Egal (/Différent) :** la valeur de l'attribut du composant doit être égale (/différente) à une valeur spécifiée par l'évaluateur ;
- **Inférieur (/Supérieur) :** la valeur (de type numérique) de l'attribut du composant graphique est inférieur (/supérieur) à celle spécifiée par l'évaluateur ;
- **Appartient à (/Différent de) :** la valeur de l'attribut du composant graphique appartient (/n'appartient pas) à un ensemble de valeurs définies par l'évaluateur ;
- **Compris entre :** la valeur, de type numérique, de l'attribut du composant graphique est comprise entre deux valeurs spécifiées par l'évaluateur.

Les valeurs spécifiées par l'évaluateur sont déduites à partir des règles ergonomiques prises en considération pour l'évaluation de la présentation statique de l'interface utilisateur. Ce module supporte un ensemble de composants graphiques pour la composition d'interfaces graphiques. Ces composants sont : bouton, fenêtre, champ de texte, case à cocher, bouton radio, onglet, boite d'image et libellé [1]. A chaque type de composants graphiques correspond des attributs. La définition porte à spécifier les valeurs de ces attributs. Ainsi l'évaluation portera sur la vérification de la conformité de la valeur (ou des valeurs) de l'attribut du composant par rapport aux valeurs spécifiées par l'évaluateur. La Figure 4.3 ci-dessous illustre la définition de la règle : « *la taille de la fonte d'écriture doit être comprise entre 14 et 16* ».

1. L'évaluateur peut ajouter de nouveaux composants graphiques à cette liste. Dans cet outil nous nous sommes contentés des composants fréquemment utilisés pour la composition d'interfaces graphiques.

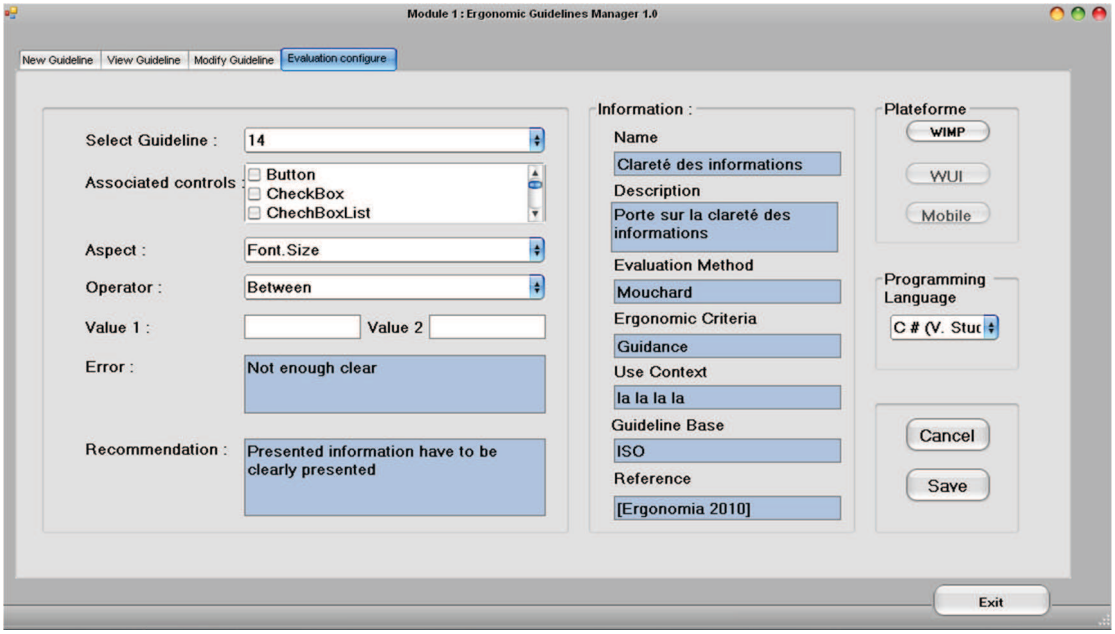



FIGURE 4.3: Gestionnaire des règles ergonomiques : interface permettant de spécifier une règle ergonomique

### 4.2.2 Module 2 : le module de capture

Ce module est chargé d'effectuer la capture des différentes données pour le processus d'évaluation. Comme le framework proposé repose sur les trois types d'évaluation : statique, dynamique et appréciation de l'utilisateur, ce module comporte trois sous-modules pour l'évaluation des *IHM* : un outil de génération de questionnaire, un parseur de données et un mouchard électronique.

#### 4.2.2.1 Outil de génération de questionnaire

L'outil de génération de questionnaire permet de spécifier un questionnaire pour l'évaluation des interfaces utilisateur. L'évaluateur crée les différentes questions. Chaque question est enregistrée sous forme de fichier XML. Afin de faciliter le processus de traitement et d'analyse des réponses aux questionnaires, nous nous restreignons à la mise en place de questions fermées. Ensuite, l'évaluateur sélectionne l'ensemble de questions pour composer le questionnaire dans le but d'évaluer l'interface utilisateur. Grâce à un script exprimée en XSLT (Aman, 2009; Williams, 2009), les fichiers XML des questions sélectionnées sont convertis sous forme de page web constituant le questionnaire, Figure 4.4. Les réponses aux questionnaires sont par la suite sauvegardées dans une base de données pour leur exploitation par le troisième module.

Une question est décrite par :
– l'identifiant unique de la question ;

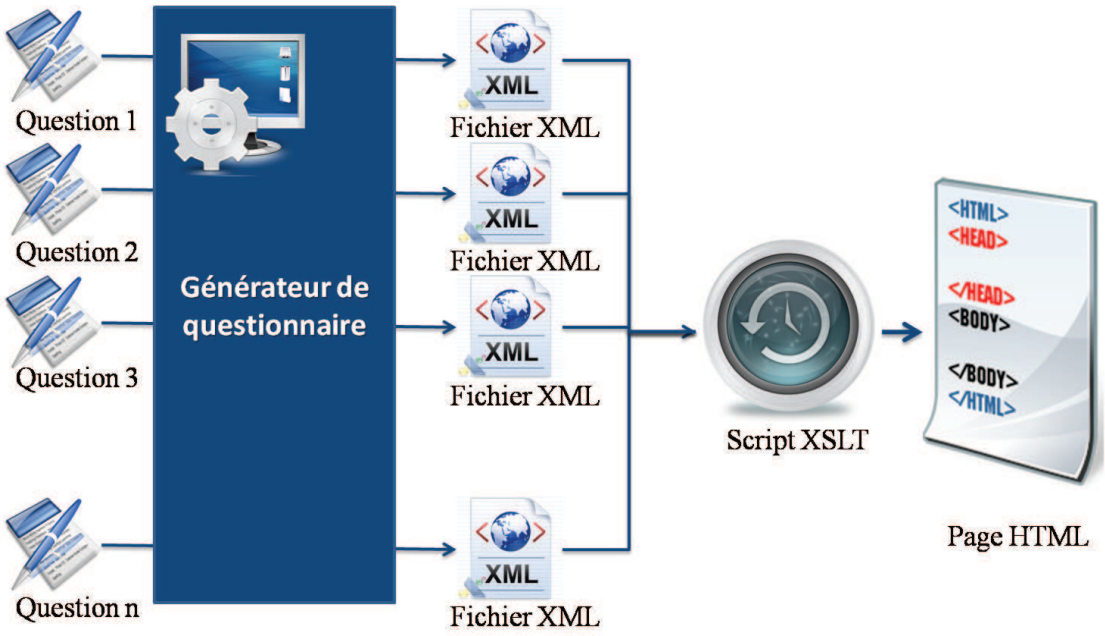



FIGURE 4.4: Principe de génération de questionnaire dédié à l'évaluation des IHM

- un titre, un descriptif de la question ;
- l'énoncé de la question ;
- les réponses de la question ;
- la dimension inspectée (utilisabilité, utilité ou accessibilité) ;
- la priorité de la question qui peut être de haute priorité, moyenne priorité ou basse priorité ;
- la source de la question ;
- le critère ergonomique. Nous avons opté pour les critères de Bastien et Scapin (Bastien et Scapin, 1993) ;
- sous-critère ergonomique et,
- le système interactif.

#### 4.2.2.2 Un parseur de données

Le parseur de données (analyseur syntaxique) permet de dégager les valeurs des attributs des composants graphiques sous forme de base de données. En entrée, l'évaluateur spécifie les composants graphiques dont il veut extraire les valeurs de ses attributs. Ensuite, il sélectionne les attributs pour chaque type de composant. Puis, il spécifie le code source de l'interface à inspecter. En sortie, les données extraites sont sauvegardées sous forme de fichier XML, Figure 4.6.

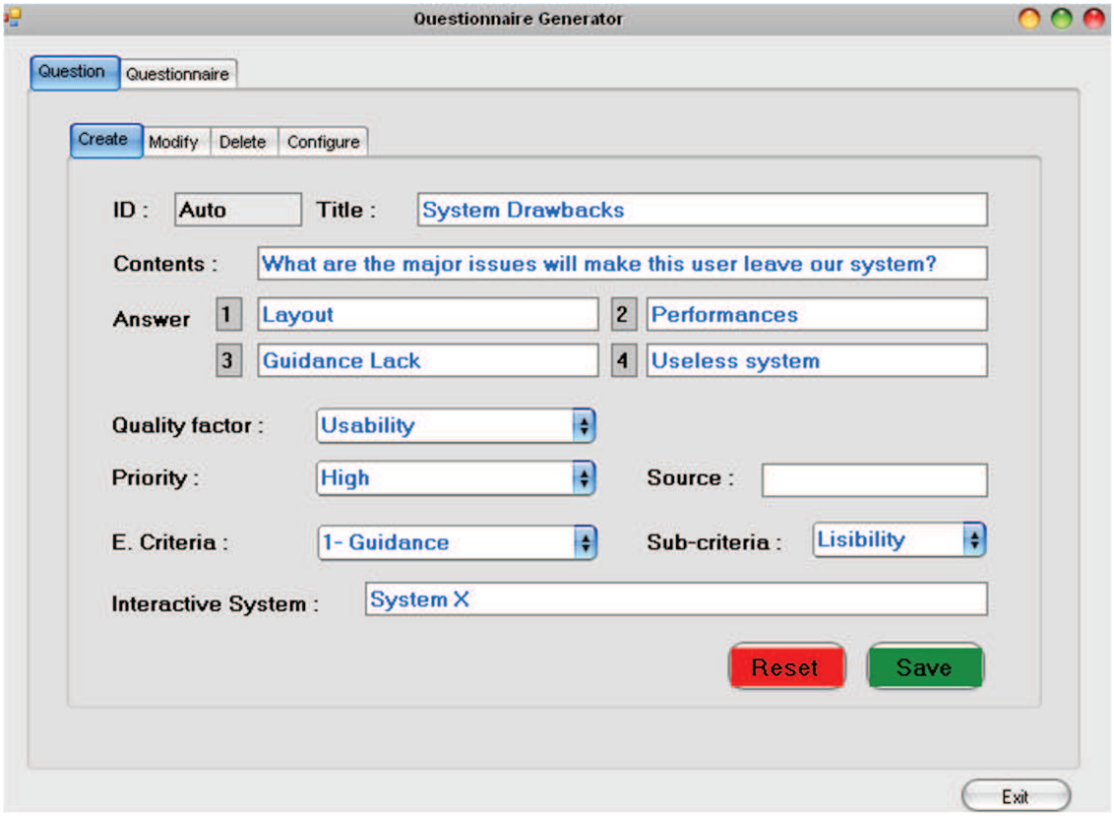



FIGURE 4.5: Interface de saisie de questions

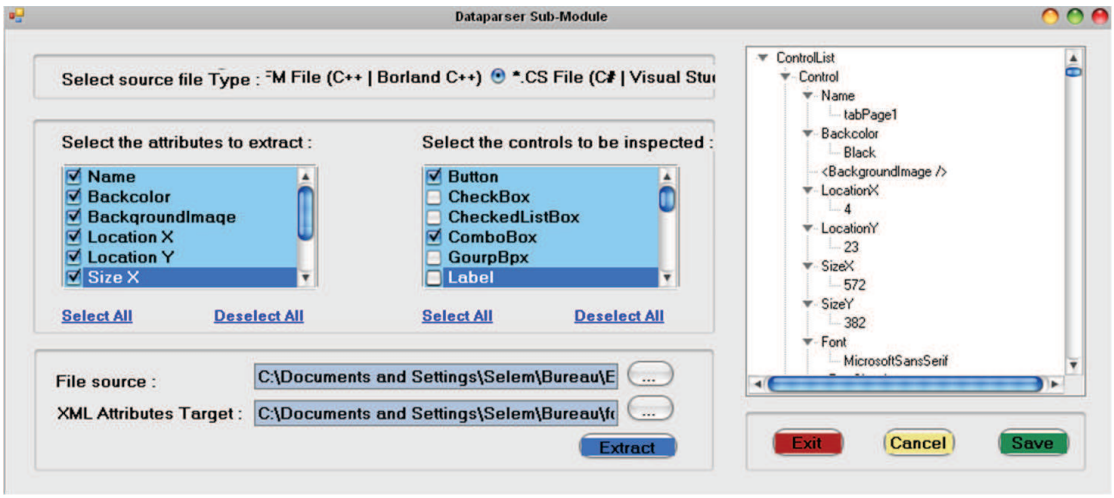



FIGURE 4.6: Interface du parseur de données

#### 4.2.2.3 Le mouchard électronique EISEval

Ce mouchard électronique est élaboré lors des travaux de la thèse de Chi Dung TRAN, Figure 4.8. Il permet de détecter et de capturer essentiellement les événements de bas niveau et de les stocker dans une base de données de type MS Access. Il permet en plus d'associer ces actions suivant des événements utilisateur. Cette classification consiste à

distinguer les différents événements portant sur les interfaces essentiellement en deux catégories, Figure 4.7 :

– **EVIU :** il s'agit des tâches exécutées à partir de l'interface utilisateur : ouverture d'une fenêtre, fermeture d'un message d'erreur, validation d'un formulaire, etc.
– **EVDI :** il consiste en l'ensemble des événements de plus bas niveau qui constituent les EVIUs. Ils traitent directement les événements générés par les dispositifs d'interaction : clic sur le bouton droit de la souris, appui sur une touche du clavier, appui sur la touche Echap, etc.

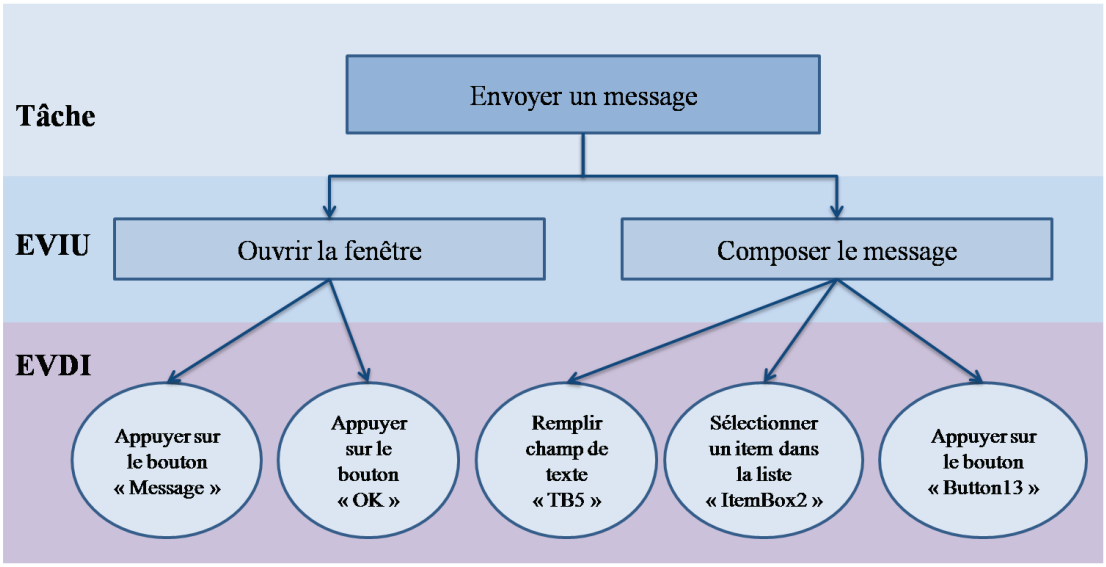



FIGURE 4.7: Exemple de décomposition d'une tâche d'EVIUs et d'EVDIs

Notons qu'une interaction est modélisée suivant le séquencement des tâches où une tâche correspond à un ou plusieurs EVIU(s). Un EVIU est lui aussi composé d'un ou plusieurs EVDI(s). La capture des événements de dispositif d'interaction se fait via l'insertion de lignes de code au niveau du code source du système interactif à évaluer. Les lecteurs intéressés peuvent consulter (Tran *et al.*, 2008; Tran, 2009).

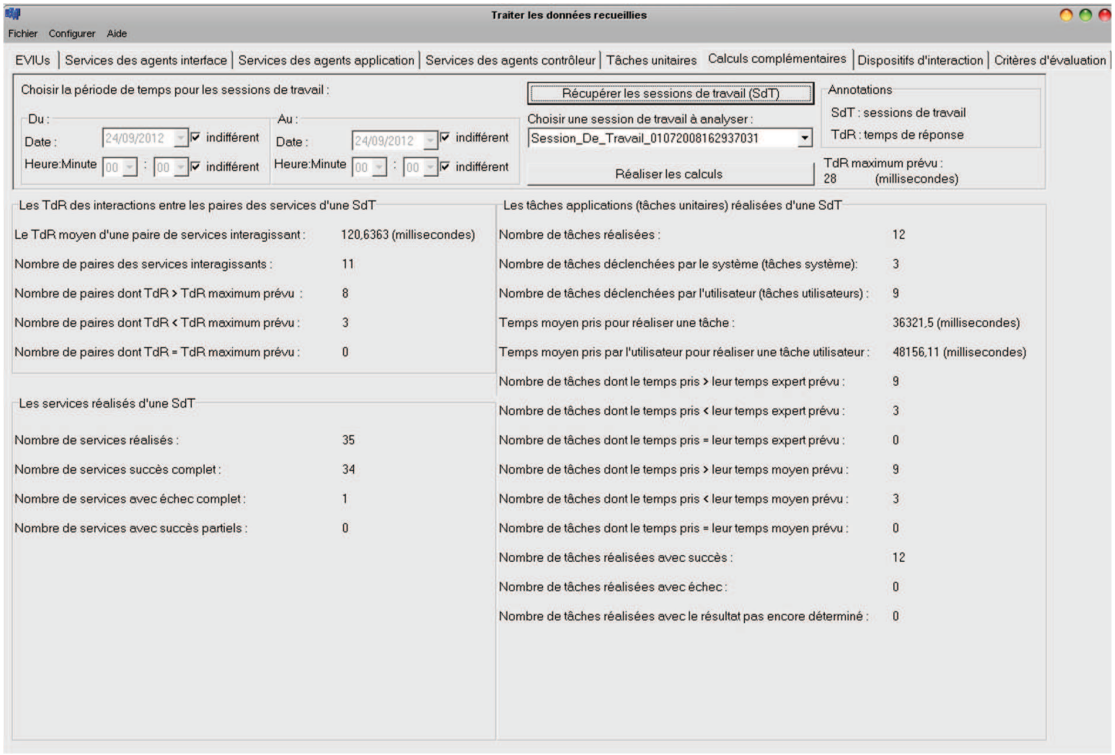



FIGURE 4.8: Aperçu de l'interface du mouchard électronique d'EISEval

Notons que le processus adopté pour le test utilisateur s'articule autour de sept étapes :

1. Identifier les utilisateurs cibles et leurs caractéristiques : les utilisateurs concernés par le test utilisateur varient d'un système à un autre suivant la population cible. Ainsi, afin d'identifier les utilisateurs cibles, il est indispensable d'identifier clairement la population cible de l'interface à évaluer. Généralement, la caractérisation des utilisateurs s'effectue suivant leur activité professionnelle, leur catégorie sociale, leur âge et leur niveau d'expertise dans le domaine associé à l'interface à évaluer.
2. Identifier les tâches utilisateur : cette phase consiste à identifier clairement les tâches à exécuter lors des sessions de test afin de bien définir les scénarios de ces tests.
3. Recruter les utilisateurs
   (a) Echantillonnage : consiste à déterminer un échantillon représentatif de la population à la quelle le système est destiné.
   (b) Recrutement dans le processus du test utilisateur : une fois un échantillon représentatif est dressé, l'évaluateur procède à l'invitation des utilisateurs pour des sessions de test. Le nombre des participants convoqués dépend de la nature du système (système destiné pour un public large ou encore un système métier).

4. Préparer le plan et le test en fonction des objectifs du test : préparer la liste des éventuelles questions à poser avant et/ou après la session de test à l'utilisateur. De plus, lors de cette phase, l'évaluateur spécifie le scénario adopté lors du test (à titre d'exemple séquencement des tâches à exécuter).
5. Préparer le matériel de test : préparer les installations matérielles de test et de recueil de données ainsi que les installations logicielles afin de procéder aux sessions de test. Il est préférable d'opter vers un test béta avec un ou deux utilisateurs pour bien vérifier le fonctionnement et la disposition des installations matérielles et logicielles.
6. Un débriefing des utilisateurs concernant le système à utiliser : consiste à introduire les objectifs du test ainsi qu'une présentation du système. De plus, les utilisateurs sont familiarisés avec le système à évaluer.
7. Elaborer les sessions de test. L'évaluateur procède à l'exécution des sessions de test avec les utilisateurs. Il est amené à veiller à ce que les utilisateurs disposent tous d'un environnement similaire. De plus, il est préférable que l'évaluateur note l'ensemble des remarques et l'appréciation de l'utilisateur lors du test.
8. Analyse des données et des observations : une fois l'évaluateur dispose de l'ensemble des données portant sur l'interaction entre l'utilisateur et le système à évaluer il procéder à l'analyse de ces données afin de générer un résultat d'évaluation. Généralement l'analyse porte tout d'abord sur la mise en place d'un ensemble de statistiques portant sur l'évaluation tel que le taux d'accomplissement des tâches, le taux d'erreur etc.

### 4.2.3 Module 3 : Moteur de l'évaluation

Ce module constitue le « *coeur* » du framework. En effet, il permet d'analyser grâce à ces quatre sous-modules les données provenant du mouchard, du questionnaire et de la description de l'interface graphique, Figure 4.9. De plus, il permet de faire la synthèse entre les résultats issus des différentes techniques. Ces sous-modules sont :

#### 4.2.3.1 L'analyseur automatique des événements utilisateur

Cet analyseur se base sur des données numériques de l'interaction telle que la durée d'exécution d'une tâche, le nombre de tâches élémentaires et le séquencement des actions élémentaires. Il permet de procéder à des comparaisons entre la moyenne des séquences d'actions élémentaires grâce au RdP *(Charfi *et al.*, 2011a). L'analyse des actions capturées s'effectue suivant trois étapes, Figure 4.10.

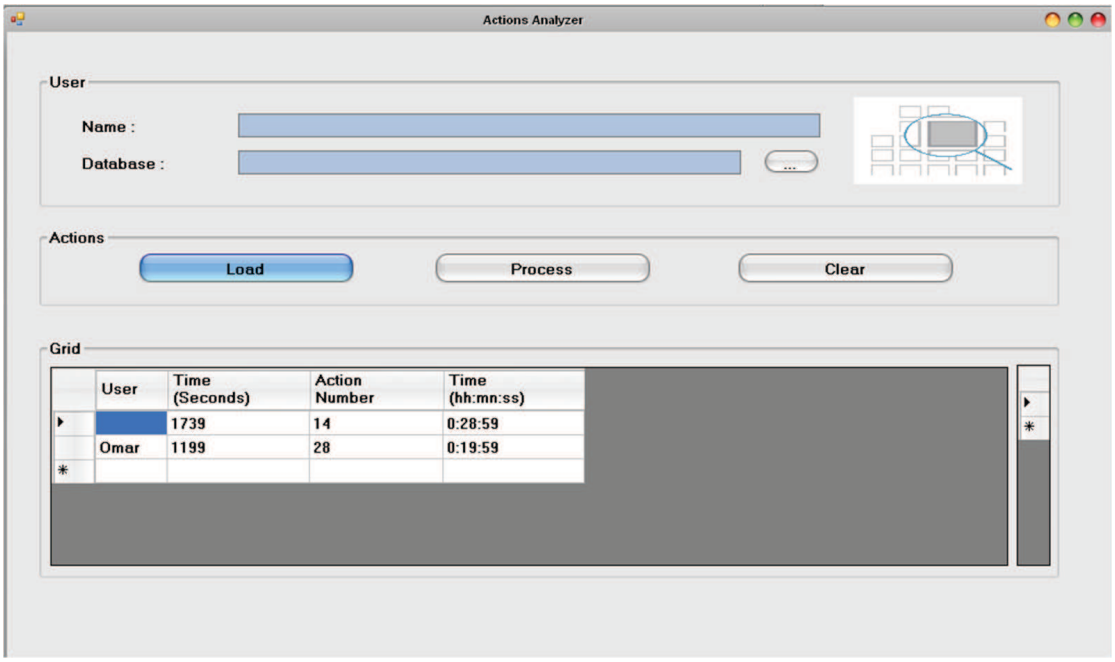



FIGURE 4.9: Aperçu de l'interface du sous-module « *Analyseur automatique des événements utilisateur* »

L'analyse des actions des différents utilisateurs consiste tout d'abord à présenter les actions exécutées et capturées entre les différents sujets et le système à évaluer sous un format de RdP : un RdP regroupant les interactions des différents utilisateurs. Ce RdP permet de déduire la séquence moyenne des différents utilisateurs afin d'exécuter une tâche. Enfin, les écarts entre les différentes séquences exécutées par les utilisateurs et la moyenne de ces séquences sont calculés. Ces derniers permettent d'aider l'évaluateur à émettre certaines critiques portant sur l'interface évaluée.

Pour celà, les interactions entre l'utilisateur et le système interactif sont structurées suivant la décomposition introduite par (Tran *et al.*, 2008) en EVDI et en EVIU. Elles sont enregistrées sous forme de fichiers XML. A partir de la définition des EVIU par des EVDI, la séquence d'interaction est traduite sous forme d'EVIU. Par la suite, un RdP regroupant les tâches exécutées par l'utilisateur est généré. Ensuite, le modèle moyen résultant de tous les modèles d'objets capturés est élaboré. Il consiste en un RdP regroupant les interactions des différents utilisateurs avec le système à évaluer durant l'évaluation expérimentale (d'autres auteurs exploitant un tel principe, par exemple (Palanque et Bastide, 1993). Ce RdP comprend les différentes séquences d'interaction ainsi que les probabilités d'effectuer chaque tâche. Cette probabilité sera calculée en fonction de la proportion des utilisateurs qui l'ont adoptée par rapport au nombre global d'utilisateurs. Les places dans ce réseau sont l'ensemble d'actions réalisées par l'utilisateur sur l'interface. Les transitions entre ces places portent des valeurs numériques de cette probabilité comme illustré en Figure 4.11.

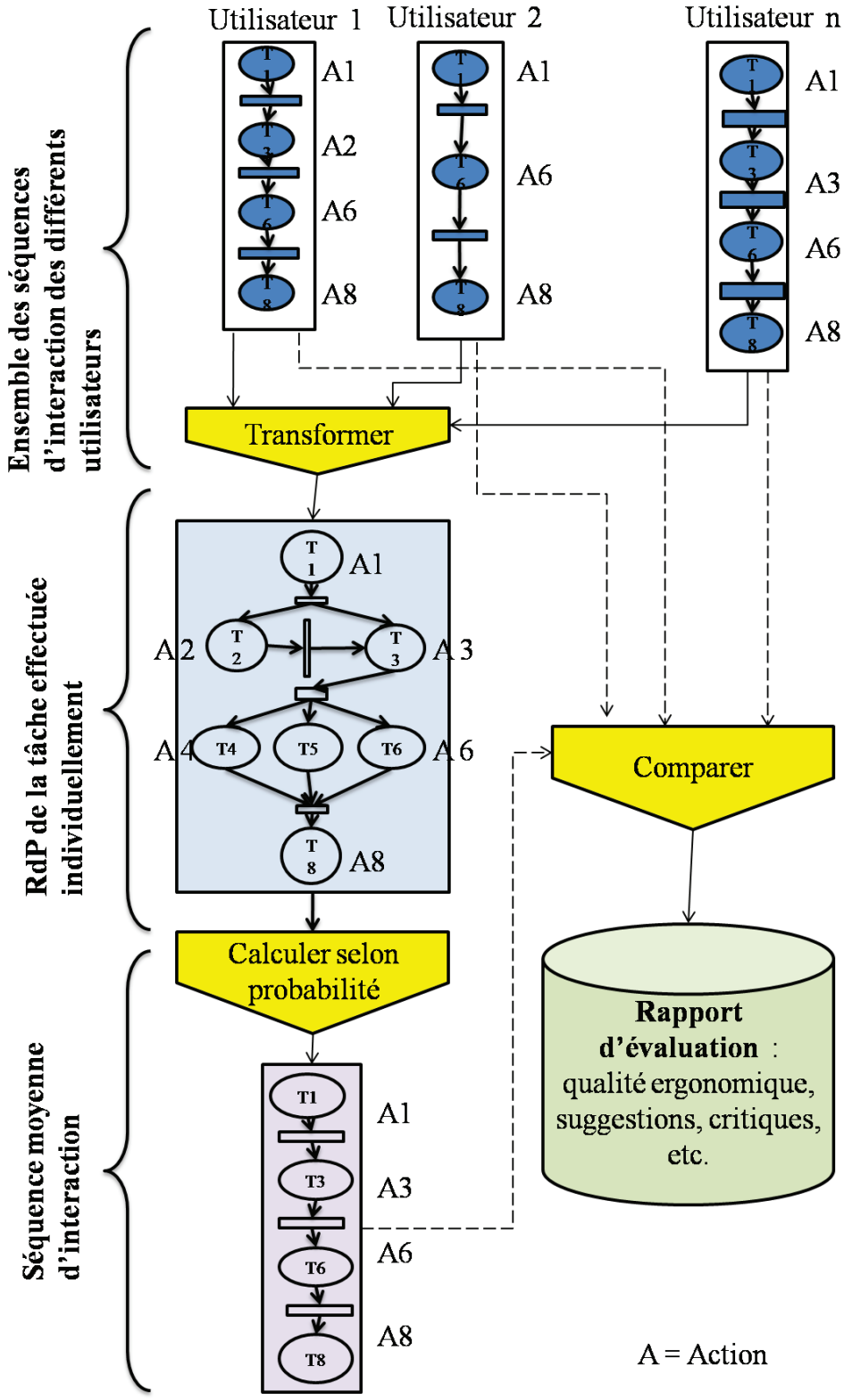



FIGURE 4.10: Principe d'analyse automatique des actions utilisateur pour l'évaluation des IHM

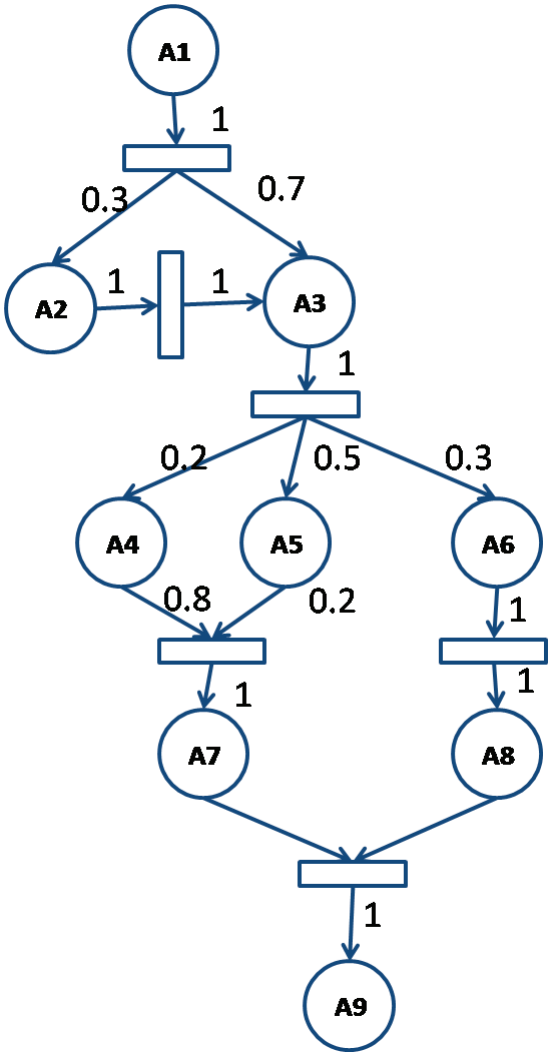



FIGURE 4.11: Exemple de RdP portant les probabilités de chaque transition

Finalement, la comparaison entre les différents modèles avec le modèle moyen permet de tirer de premières conclusions sur la qualité de l'interface évaluée. La comparaison s'effectue suivant trois niveaux : (1) nombre d'actions, (2) temps d'exécution, (3) écart entre différentes séquences d'action effectuée par l'utilisateur. Si les valeurs de l'écart ne sont pas importantes, nous pouvons affirmer que les utilisateurs ont procédé suivant le même état d'esprit et que le système a bien présenté les différentes informations à l'utilisateur. Des différences sont tolérées du fait que les utilisateurs ne possèdent pas tous les mêmes capacités (de mémorisation, concentration, etc.). Dans ce cas, nous pouvons déduire que les utilisateurs ont opté pour les mêmes séquences de tâche dans le scénario d'expérimentation et que les données sont bien présentées sur cette interface. Dans le cas contraire, nous pouvons mettre en évidence suivant la nature de l'écart, des problèmes rencontrés par l'utilisateur lors de l'interaction avec le système :

– écart en termes de temps d'exécution : il est possible de conclure que l'utilisateur a éprouvé des difficultés au niveau de la lecture et du traitement de l'information. Si

certains utilisateurs ont présenté une lenteur à manipuler le système à évaluer, ceci peut suggérer des problèmes au niveau des critères ergonomiques : signifiance des codes et dénominations [2] et charge de travail [3] (au sens de (Bastien et Scapin, 1993)).

– écart entre le nombre de transitions pour chaque séquence d'utilisation, il est possible de déduire qu'un utilisateur a procédé à des manipulations inutiles. Ceci touche aux critères de guidage, homogénéité/cohérence et de prévention contre les erreurs (au sens de (Bastien et Scapin, 1993)).
– écarts importants dans les différentes séquences d'utilisation pour un même scénario, il est possible de déduire que certains aspects dans l'interface à évaluer sont à revoir. Ces critères sont essentiellement le guidage et la prévention contre les erreurs (au sens de (Bastien et Scapin, 1993)).

Notons que nous avons calculé les valeurs limites d'un écart à partir d'une population de 100 utilisateurs interagissant avec un système de billetterie automatique. Les séquences d'interactions sont capturées à partir d'un mouchard électronique. Les résultats générés à partir d'une base de données, dans laquelle le mouchard stocke les différentes actions, montrent qu'il y a en moyenne un écart de 17 % en temps d'exécution et un écart de 14 % en nombre d'actions. Cet écart est calculé à partir de la courbe illustrée dans la Figure 4.12.

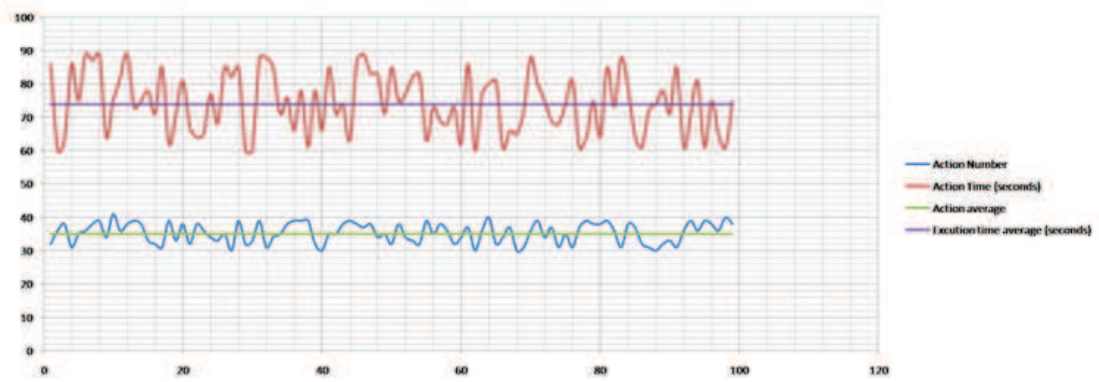



FIGURE 4.12: Courbe illustrant le temps d'exécution et le nombre de tâche pour le calcul de la moyenne d'écart

2. La signifiance des codes et dénominations est un critère ergonomique suivant les critères proposés par Scapin et Bastien. Il porte sur l'adéquation entre l'objet ou l'information affichée, et son référent (Bastien et Scapin, 1993).

3. La charge de travail est un critère ergonomique suivant les critères proposés par Scapin et Bastien. Il consiste à réduire la charge de travail perspective ou mnésique des utilisateurs (Bastien et Scapin, 1993).

#### 4.2.3.2 L'analyseur de questionnaires

L'analyseur des questionnaires permet de synthétiser les réponses des utilisateurs aux questionnaires afin d'assister l'évaluateur pour détecter les éventuels problèmes d'utilisabilité et d'utilité du système interactif, Figure 4.13. Il fournit en sortie un fichier Excel. L'analyse des questions s'effectue via la synthèse de ces dernières sous forme de graphes permettant d'identifier les problèmes d'utilisabilité, d'utilité et d'accessibilité de l'interface.

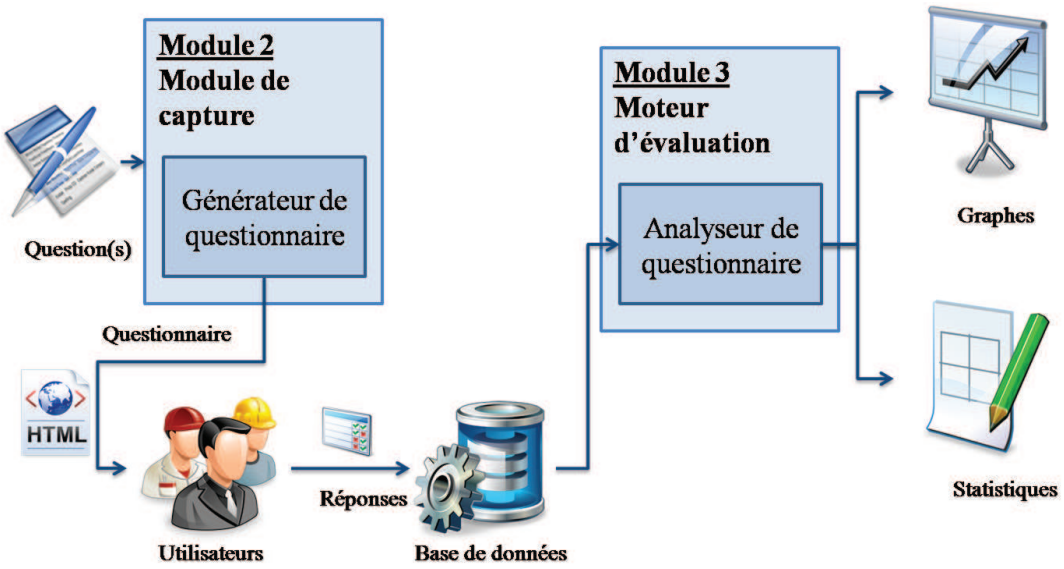



FIGURE 4.13: Principe de l'exploitation du questionnaire pour l'évaluation des IHM

#### 4.2.3.3 Analyseur de l'interface graphique

L'analyseur de l'interface graphique permet de détecter des problèmes de présentation de l'interface graphique suivant les règles ergonomiques spécifiées par l'évaluateur pour le processus de l'évaluation, Figure 4.14. Nous pouvons citer à titre d'exemples des problèmes de lisibilité des informations et d'affichage des données.

Ce sous-module opère comme suit. En se référant à une base de données des règles et manuels ergonomiques, l'évaluateur identifie des critères ergonomiques qu'il estime essentiels et fondamentaux pour l'évaluation des *IHM*. Ces règles sont traduites via le premier module de ce framework sous forme de fichier XML. Ensuite, le fichier source (dfm [4] ou bien cs [5].) est chargé dans le sous-module de parseur de code pour dégager les valeurs nécessaires pour l'évaluation. Ainsi en entrée à ce sous-module, l'évaluateur fournit les règles à inspecter ainsi que les valeurs des attributs des composants de l'interface

4. les fichiers d'extension dfm sont associés aux environnements de conception Delphi.
5. Il s'agit d'extension de fichier source d'application développée sous Visual C#

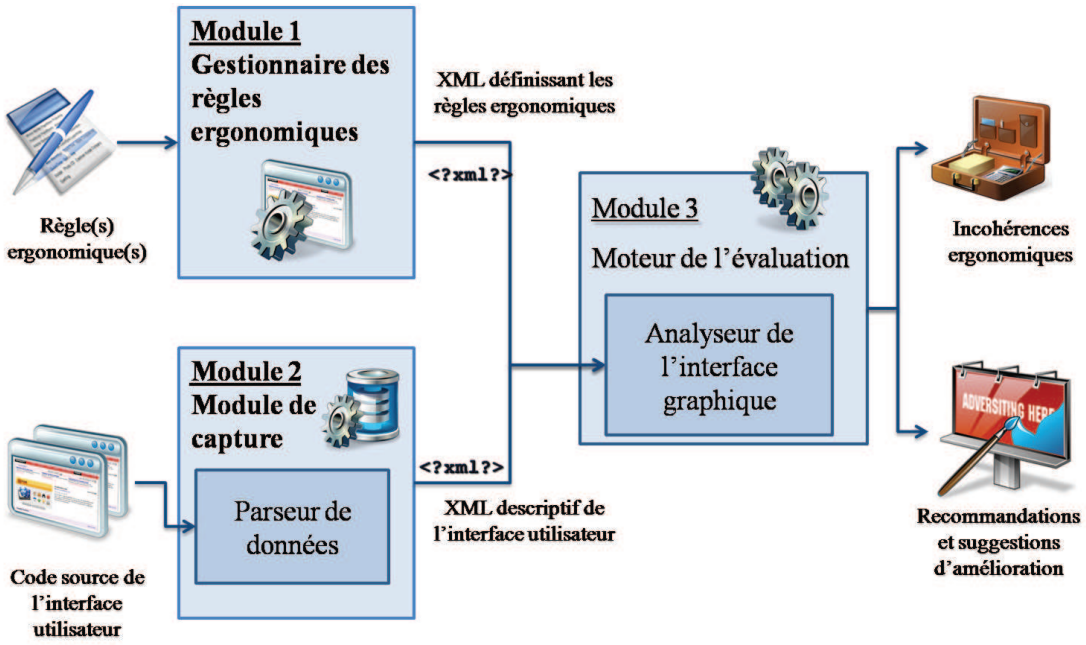



FIGURE 4.14: Principe de l'exploitation de l'inspection de la qualité ergonomique pour l'évaluation des IHM

à évaluer. Ce sous-module effectue des comparaisons entre ces deux entrées et génère en sortie les résultats de telle comparaison. Cette comparaison comprend l'ensemble des incohérences ergonomiques de l'interface par rapport à l'ensemble des règles définies. Le rapport dévaluation comprend ainsi les résultats de la comparaison qui consistent à identifier les incohérences ergonomiques détectées dans l'interface par rapport à l'ensemble de règles sélectionnées par l'évaluation. Ce rapport est proposé suivant deux formats XML et PDF, Figure 4.15.

#### 4.2.3.4 L'interpréteur des données

Ce sous-module permet de synthétiser entre les résultats des trois autres sous-modules cités ci-dessus. Il consiste à grouper le résultat d'évaluation suivant les tâches exécutées à travers le système interactif évalué. A chaque tâche sont associés trois coefficients compris entre 0 et 1. Le premier correspond au ratio des utilisateurs qui ont exécuté la tâche avec succès (par exemple 0,5 si la moitié des utilisateurs sont parvenus à exécuter la tâches avec succès). Le second coefficient correspond au ratio des réponses positives des utilisateurs portant sur cette tâche. Enfin le troisième aspect comporte la proportion de composants graphiques utilisés lors de l'exécution de cette tâche et qui sont conformes aux règles ergonomiques spécifiées lors du début de processus d'évaluation.

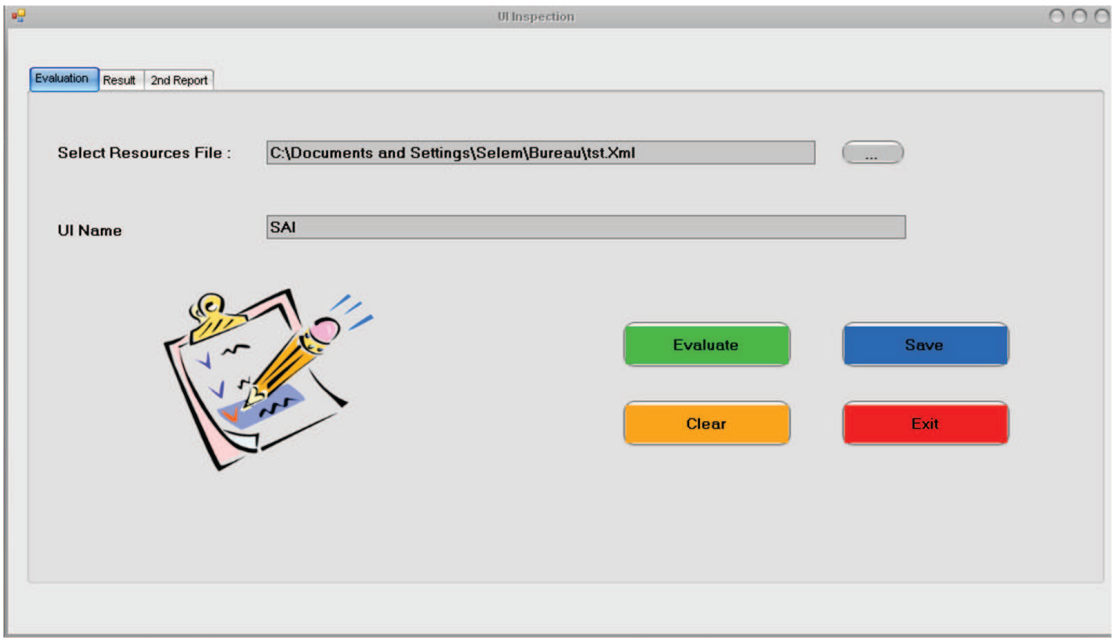



FIGURE 4.15: Interface de l'analyseur de l'interface graphique

| Tâche | Coefficient Mouchard Electronique (C1) | Coefficient Questionnaire (C2) | Coefficient Inspection Ergonomique (C3) |
|---|---|---|---|
| T1 | 0.90 | 0.50 | 0.20 |
| T2 | 1.00 | 0.80 | 0.60 |
| T3 | 0.50 | 0.90 | 0.36 |

TABLE 4.1: Exemple de données pour l'évaluation des IHM

Lors de l'évaluation, nous nous basons sur la classification de Nielsen des erreurs dans une interface (Nielsen, 1993). Ce tableau permet de dresser le degré des erreurs détectées dans une interface suivant la logique suivante, Tableau 4.1 :

– entre 1.0 et 0.9 : cosmétique ;
– entre 0.9 et 0.8 : mineure ;
– entre 0.8 et 0.7 : moyenne ;
– entre 0.7 et 0.5 : majeure et,
– entre 0.5 et 0.0 : critique.

Lors de l'évaluation, le coefficient issu depuis le mouchard électronique est estimé comme étant le coefficient ayant le plus de poids. En effet, nous estimons que la capture et l'analyse des interactions est la technique la plus fiable parmi les trois utilisées lors de l'évaluation. En effet, elle se base sur des données numériques quantitatives. De plus, elle ne requiert aucun jugement subjectif de la part de l'évaluateur. L'inspection de la qualité ergonomique et le questionnaire proposent des résultats complémentaires au mouchard. Ainsi la logique de décision adoptée est :

– C1*C2*C3 =1 : l'interface ne présente aucun problème d'utilisation lors de l'exécution de la tâche. Nous pouvons confirmer ainsi qu'elle est d'excellente qualité (suivant

la facilité d'interaction, la présentation statique de l'interface et l'appréciation de l'utilisateur).

- C1> 0.9 : L'interface évaluée est d'une excellente qualité. Néanmoins, nous retrouvons des problèmes d'ordre cosmétique.
- 0.8 < C1 <0.9 : Très bonne interface coté *IHM*. L'évaluation a permis d'identifier des problèmes mineurs.
  - Si C2 = 1 : Si l'évaluateur va procéder à une deuxième évaluation, il est sollicité pour prendre en considération d'avantage de règles ergonomiques.
  - Si C2< 1 : correction de l'interface
  - Si C3<0.5 : problèmes d'acceptabilité de l'interface par l'utilisateur
- 0.7 < C1 <0.8 : interface de bonne qualité avec certains problèmes de degrés moyen à rectifier
- 0.5 < C1 <0.7 : interface de moyenne qualité avec des corrections sont à effectuer relevant à des erreurs de conception de degrés majeur,
- C1 < 0.5 : interface qui présente beaucoup de problème d'utilité et d'utilisabilité et qui requiert une correction : les problèmes détectés sont d'ordre critique.

Ainsi ce sous-module présente le résultat de la synthèse de l'évaluation dans un fichier XML : qualité de l'interface, erreurs de conception de l'interface et les recommandations d'amélioration.

### 4.2.4 Module 4 : Outil de génération de rapports

Ce module est chargé de présenter le résultat de l'évaluation suivant un rapport bien compréhensible et exploitable par l'évaluateur dans le but d'améliorer la qualité er l'interface présentée. Dans la littérature, il n'existe pas beaucoup de formalismes permettant de présenter le rapport d'évaluation. Le rapport proposé peut être exposé suivant trois formats : PDF, HTML et TXT, Figure 4.16. Ce rapport contient essentiellement le bilan de l'évaluation basée sur les trois techniques d'évaluation.

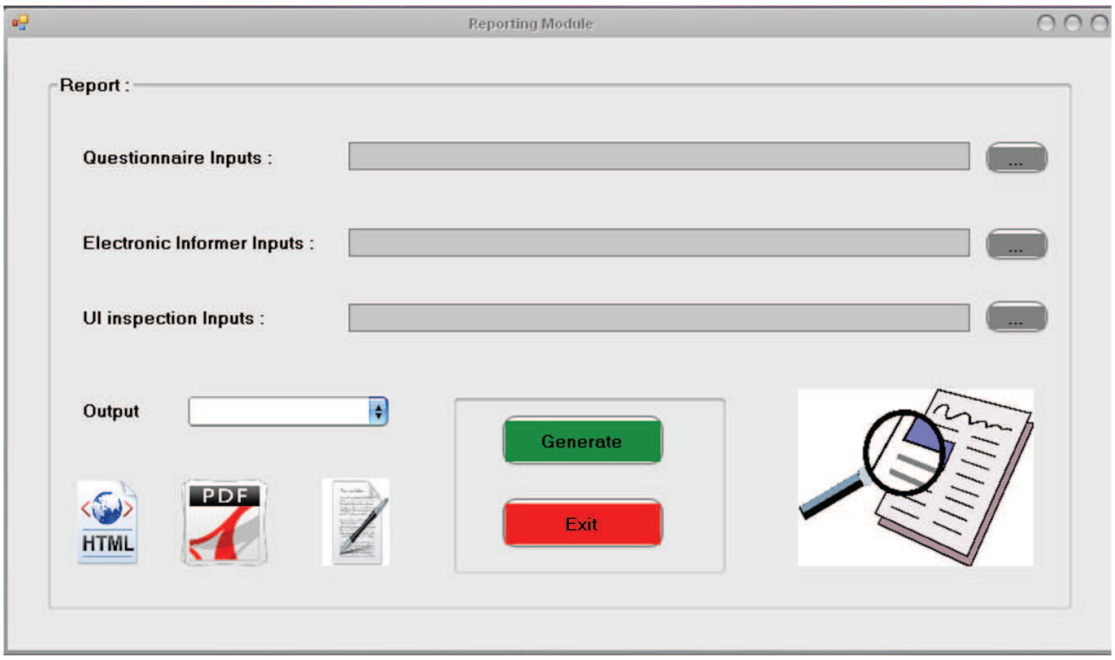



FIGURE 4.16: Interface de l'outil de génération de rapport d'évaluation

## 4.3 Processus d'évaluation

Lors de cette section nous allons introduire le processus d'évaluation d'une *IHM* à travers le framework *RITA*. Ce processus requiert la participation essentiellement de trois acteurs : l'évaluateur (pour gérer et assister le processus d'évaluation), un expert en ergonomie logicielle (pour sélectionner les règles ergonomiques) et des utilisateurs (pour participer aux sessions de test). Le processus d'évaluation proposé s'articule essentiellement autour de trois phases. La première phase consiste dans la préparation des données pour le processus de l'évaluation. La seconde quant à elle est l'évaluation. Ensuite la troisième phase comporte l'analyse et la synthèse des données issues de l'évaluation.

### 4.3.1 Phase de pré-évaluation

Cette phase consiste à préparer les données pour l'évaluation et à mettre en place les dispositifs permettant la capture des données pour l'évaluation. Elle s'effectue comme mentionné dans Figure 4.17.

Tout d'abord l'évaluateur spécifie l'objectif de l'évaluation prévue. Cette spécification consiste à identifier les aspects à prendre en considération lors de l'évaluation (la présentation statique de l'interface doit être conforme à certaines normes, identifier les problèmes que l'utilisateur peut rencontrer dans l'interface lors de l'interaction, valider la qualité de l'interface, etc.). Les aspects sont déterminés essentiellement suivant le contexte de l'utilisation du système interactif (domaine d'application, profil utilisateur,

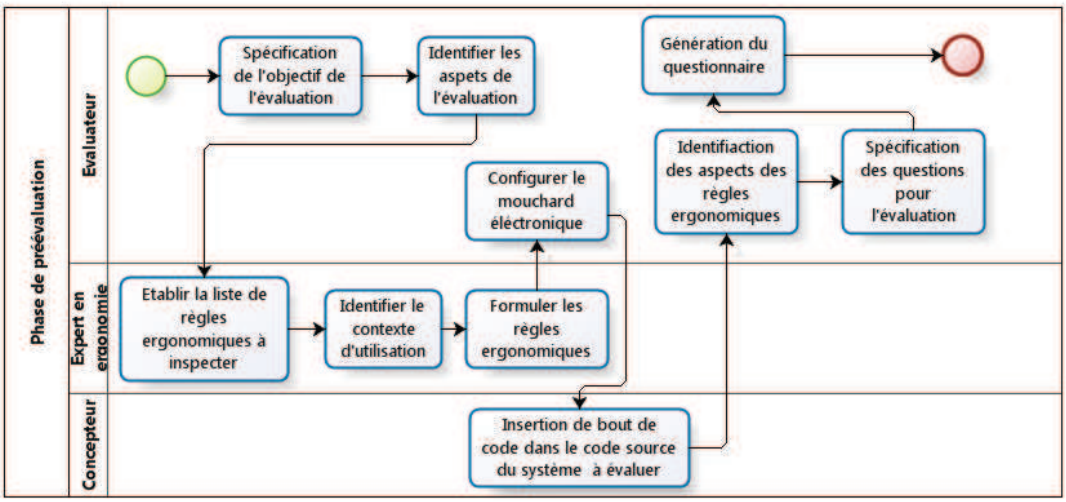



FIGURE 4.17: Processus de pré-évaluation modélisé avec la notation BPMN

type d'interface, etc.). Ensuite, avec l'aide d'un expert en ergonomie logicielle, il établie une liste de règles ergonomiques à inspecter pour la conformité dans l'interface utilisateur. Ces règles sont par la suite exprimées sous forme de fichiers XML à l'aide du premier module du framework *RITA*. Puis, l'évaluateur configure le mouchard électronique EISEval pour la capture des actions utilisateurs lors des sessions d'interactions entre les utilisateurs et le système interactif (spécification des tâches à exécuter, association des actions élémentaires aux différentes tâches, etc.). Afin de capturer les actions utilisateur, des lignes de code doivent être intégrées au niveau du code source du système à évaluer. Par la suite, l'évaluateur recense les aspects visés lors de l'évaluation et qui ne sont pas supportés pour l'évaluation par le mouchard et les règles ergonomiques. Ces aspects servent à établir des questions pour interroger l'utilisateur sur ces aspects. Les questions sont créées par le sous-module de génération de questionnaire du module capture de données.

### 4.3.2 Phase d'évaluation

Cette phase consiste à extraire les données nécessaires pour l'évaluation et les analyser en vue d'identifier les problèmes d'utilité et utilisabilité de l'interface utilisateur. Cette phase commence tout d'abord par l'extraction des données. Cette opération est organisée en trois étapes. La première consiste à extraire les données nécessaires pour la validation de l'interface par rapport aux règles ergonomiques sélectionnées par l'évaluateur. Elle consiste à parcourir le code source du système interactif et à en extraire ces données dans des fichiers XML. La seconde étape consiste à mener des sessions de tests avec des utilisateurs et d'enregistrer les interactions des utilisateurs avec l'interface dans le but de les étudier. Ces sessions consistent à élaborer tout d'abord des scénarios d'évaluation.

Le but de ces scénarios est d'exécuter le maximum de tâches possibles avec le système afin d'identifier le maximum de problèmes éventuels d'utilisabilité et d'utilité dans l'interface. Ensuite, l'évaluateur doit sélectionner un ensemble d'utilisateurs pour la session de test, Figure 4.18. Lors de l'évaluation expérimentale, il est important de faire appel à des échantillons représentatifs des profils des utilisateurs censés utiliser le système. Ensuite l'évaluateur doit s'assurer de fournir l'environnement adéquat pour l'interaction en situation très similaire aux conditions normales de l'utilisation du système interactif. Par la suite, l'évaluateur mène l'expérimentation et parallèlement le mouchard capture et enregistre ces données.

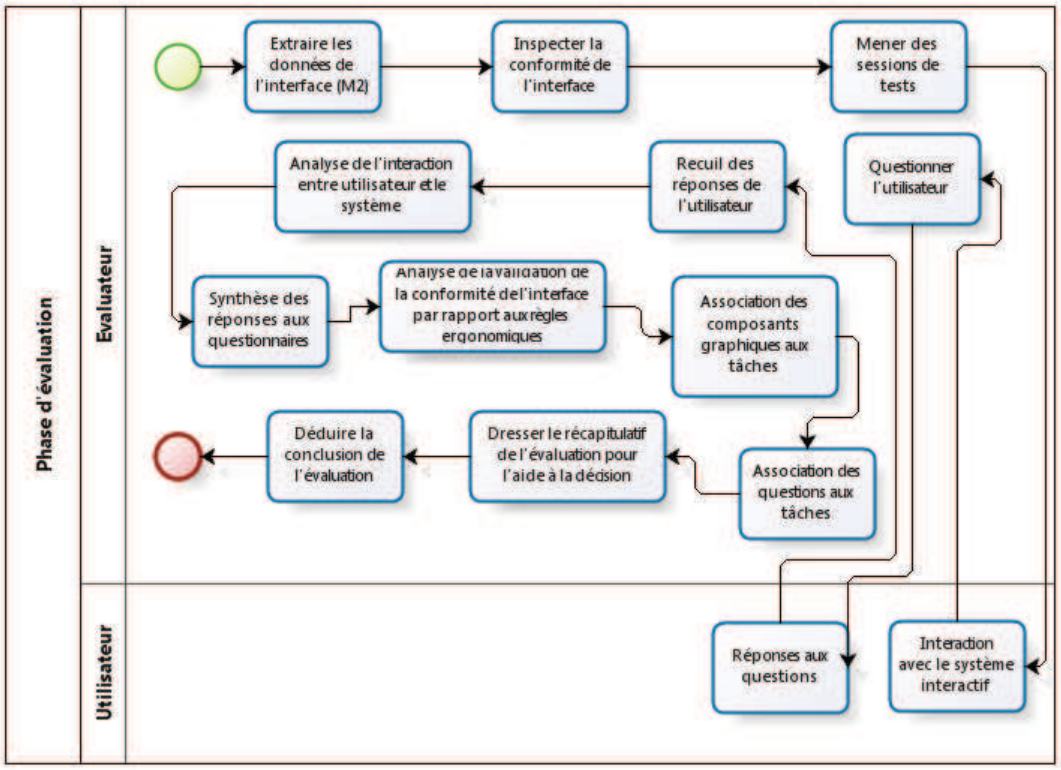



FIGURE 4.18: Processus de l'évaluation modélisé avec la notation BPMN

A la fin de l'expérimentation, l'évaluateur propose un questionnaire aux utilisateurs pour avoir leur avis concernant le système interactif ainsi que leurs réponses concernant les aspects pris en considération lors de l'évaluation et qui sont impossible à inspecter via l'inspection de la qualité ergonomique et le mouchard électronique. Ce questionnaire concerne le troisième point de l'extraction des données de l'évaluation. Une fois les utilisateurs ont répondu au questionnaire l'évaluateur dispose de toutes les données nécessaires pour mener l'évaluation. Les données issues de l'extraction des valeurs des attributs des composants de l'interface graphique sont comparées par rapport aux recommandations issues des règles ergonomiques sélectionnées lors de pré-évaluation de l'interface pour dégager les aspects à améliorer lors de présentation statique de l'interface. Ensuite, les

données capturées par le mouchard sont analysées pour déduire des conclusions sur le taux de succès de l'interaction des utilisateurs avec le système interactif. Les statistiques proposées sont des premières conclusions de l'évaluation. L'évaluateur établie ensuite une association entre les différentes composants graphiques (et les règles associées pour chaque composant) aux tâches exécutées. Le même travail est élaboré pour les questionnaires. Ensuite un tableau est dressé pour dégager les conclusions portant sur l'évaluation du système interactif.

### 4.3.3 Phase de post-évaluation

Cette phase consiste à générer un rapport qui regroupe le bilan de l'évaluation. Ce bilan est constitué de :

- Rapport des problèmes détectés sur le système interactif ;
- Les suggestions et les recommandations d'améliorations à apporter au système interactif et,
- Un compte rendu sur le déroulement de l'évaluation, ce compte rendu englobe les problèmes rencontrés lors du processus de l'évaluation.

Une fois ce rapport généré, l'évaluateur ou le concepteur procède à appliquer les différentes suggestions et recommandations d'améliorations de l'interface sur le système interactif afin de remédier aux différents problèmes d'utilité et d'utilisabilité détectés dans ce système.

## 4.4 Discussion

Lors de la première partie de ce mémoire nous avons étudié les différents techniques et outils d'évaluation des systèmes interactifs. Suite à l'étude de ces outils, nous nous sommes aperçus des limites des outils existants pour l'évaluation des systèmes interactifs. Les limites des ces outils existants ont permis la proposition et l'élaboration du framework lors de ce présent chapitre. Ce framework est muni d'une architecture modulaire permettant sa configuration pour l'évaluation des différents systèmes interactifs. De telle architecture peut s'intégrer dans d'autre environnement ou plateforme d'évaluation (que ce soit partiellement ou totalement). De plus, l'évaluateur peut se contenter d'utiliser quelques outils du framework proposé pour évaluer les systèmes interactifs suivant ses besoins, contraintes et les spécifications du processus d'évaluation.

La première limite identifiée dans l'étude du besoin concerne le codage en dur des règles ergonomiques dans le moteur d'évaluation dans la plupart des outils présentés ayant la

spécificité de se baser sur des connaissances ergonomiques pour l'évaluation des systèmes interactifs. En effet, dans le framework proposé nous avons opté pour une modélisation des règles sous forme de fichiers XML. Ces fichiers sont externes au moteur d'évaluation et par la suite l'évaluateur (un expert en ergonomie) peut modifier l'ensemble des règles prises en considération lors de l'inspection de la qualité ergonomique de l'interface utilisateur. La modification peut s'effectuer en ajoutant de nouvelles règles, supprimer ou modifier les règles déjà existantes via un éditeur dédié à cet usage. Cet éditeur est baptisé « *Ergonomic Guideline Modeller* ». Cependant l'éditeur proposé ne permet que de modéliser les règles une à une et ne permet pas de modéliser plusieurs règles suivant une seule structure. De plus, l'éditeur proposé ne permet pas de modéliser toutes les règles. Certaines règles telles que celles concernant la clarté des messages d'erreurs ne sont pas supportées par l'éditeur proposé.

D'autres limites ont été identifiées concernant le choix de la technique adéquate pour l'évaluation des systèmes interactifs. En effet, il existe de nombreuses techniques qui ont chacune des spécificités et des aspects à évaluer différents des autres. Les résultats d'évaluation peuvent converger suivant la technique utilisée. Par exemple, une interface peut être validée par rapport à un ensemble de règles ergonomiques alors que son exploitation est assez difficile par l'utilisateur. Dans ce framework, nous avons opté pour se baser sur trois différentes techniques pour l'évaluation des systèmes interactifs. Ces techniques sont le mouchard électronique, la validation de l'interface par rapport à un ensemble de règles ergonomiques et le questionnaire. Ainsi, le framework proposé considère l'évaluation suivant nombreux aspects. D'une part, il se focalise sur la présentation statique des informations et composants graphiques mis à la disposition de l'utilisateur. D'autre part, il s'intéresse à l'interaction avec l'utilisateur (temps de réponse, sequencement des actions pour l'exécution d'une tâche donnée, etc.). Ainsi, l'évaluation prend en considération les performances de l'utilisateur lors de l'interaction avec l'utilisateur. De plus, le framework prend en considération la satisfaction de l'utilisateur de l'interface évaluée. Ainsi, le processus d'évaluation se base sur différents aspects pour l'évaluation des systèmes interactifs et permet de générer un résultat issu de trois techniques différentes d'évaluation. Ceci justifie ainsi le recours à ces techniques d'évaluation ; ce choix est accentué par la volonté de couvrir un spectre plus large lors de l'évaluation des *IHM*. Cependant, l'utilisation de ces techniques révèle quelques difficultés d'exploitation. En effet, le mouchard utilisé requiert l'ajout de ligne de code au niveau du code source du système évalué. Il n'est pas évident que l'évaluateur puisse disposer du code du système à évaluer. Quant à l'inspection des règles ergonomiques, l'évaluateur doit procéder par l'évaluation des *IHM* interface par interface.

Concernant le rapport d'évaluation, les résultats d'évaluation sont proposés à l'évaluateur suivant plusieurs formats. De plus, il est composé de détails afin de lui permettre

plus de flexibilité lors de l'exploitation de ces résultats. Le rapport contient essentiellement trois types de résultats. Le premier consiste dans l'ensemble des problèmes d'utilité et d'utilisabilité détecté dans l'interface. Ensuite, une liste de recommandations et de suggestions d'évaluation est proposée afin de remédier aux incohérences présentées au niveau du système interactif. Puis, l'évaluateur dispose des informations portant sur le déroulement du processus d'évaluation (fichier illisible, trame tronquée reçue par le mouchard, etc.).

Au niveau de l'automatisation du framework proposé, la phase de capture est automatisée tout comme l'analyse des données d'évaluation. La phase de critique reste partiellement automatique et requiert l'intervention de l'évaluateur pour conclure sur le résultat de l'évaluation. Il existe des difficultés pour automatiser les critiques du fait qu'au niveau de l'évaluation des *IHM* , tout comme le génie logiciel, il n'existe pas de solution universelle ou encore de démarche canonique. Ainsi, il est difficile, voire même impossible, de procéder à cette phase automatiquement sans aucune intervention des opérateurs humains (Nielsen, 1993). L'utilisation de ce framework pour l'évaluation requiert la présence d'opérateurs humains (évaluateur, utilisateur et concepteur) et certaines compétences (en ergonomie, programmation et évaluation).

Chapitre 5

# Contribution 2 : Composants graphiques orientés évaluation des interfaces utilisateur

## Introduction

> « *J'ai toujours rêvé que mon ordinateur soit aussi simple à utiliser que mon téléphone. Ce rêve est devenu réalité : je ne comprends plus comment utiliser mon téléphone.* »
>
> Bjarne Stroustrap [1]

Cette citation illustre la complexité qu'ont atteint les différents systèmes interactifs sur différentes plateformes au niveau de leur utilisation et leur exploitation (PC, table tactile, PDA, téléphone portable, etc.). En effet, l'évolution de l'informatique a engendré de nouveaux challenges concernant la promotion de techniques et de méthodes permettant de concevoir des interfaces graphiques utiles et utilisables (Koua et Kraak, 2004). Lors du second et du troisième chapitre nous nous sommes focalisé sur certains de ces travaux qui consistent essentiellement en des outils d'évaluation des interfaces utilisateur. Bien que ces outils soient nombreux, nous relevons tout de même des difficultés liés à l'évaluation des *IU*. Dans ce chapitre nous allons proposer une contribution pour l'évaluation des interfaces utilisateur dans le but de remédier aux limites des outils existants. Cette contribution s'articule autour de composants graphiques pour la

1. Bjarne Stroustrap est un professeur en sciences informatiques. Il est l'un des auteurs du langage de programmation C++.

composition d'interfaces utilisateur. Ces composants ont la spécificité de supporter la phase d'évaluation. Ainsi, cette contribution consiste à coupler entre la conception des interfaces utilisateur et leur évaluation ; c'est le cas de l'évaluation précoce. Ce chapitre est structuré suivant cinq sections. Lors de la première section nous allons présenter informellement la contribution proposée : la motivation de sa réalisation et son concept. Ensuite, nous présenterons les deux familles de composants proposés : les composants dédiés à l'inspection de l'utilisabilité et ceux dédiés au test utilisateur. Par la suite, la quatrième section introduira le processus d'évaluation adopté pour l'évaluation des interfaces utilisateur avec la contribution proposée. Enfin, ce chapitre sera conclu par une discussion permettant de situer la contribution proposée par rapport à l'existant et de spécifier ses limites pour l'évaluation des interfaces utilisateur.

## 5.1 Présentation générale de la contribution

### 5.1.1 Motivation

Comme mentionné en chapitres 2 et 3, de nombreux outils dédiés pour l'évaluation des interfaces utilisateur existent. Néanmoins, ces outils présentent des limites. Nous citons essentiellement la difficulté et la complexité du processus d'évaluation adopté pour identifier les problèmes relatifs à l'utilité et à l'utilisabilité des interfaces utilisateur (Hearst, 2009). Ainsi, la phase d'évaluation est, dans la majorité des cas, négligée par les concepteurs (Grislin et Kolski, 1996). De plus, nous ne trouvons pas assez d'outils procédant à une évaluation précoce des *IU*. A noter que l'évaluation précoce repose sur trois catégories de techniques d'évaluation : l'inspection heuristique, l'application des principes de l'utilisabilité et les tests d'utilisabilité sur les prototypes (Hvannberg *et al.*, 2007). Généralement afin de procéder à une évaluation précoce, les évaluateurs procèdent à la technique de prototypage (Buxton, 2007). En effet, la mise en place de prototype est rapide et donc non coûteuse. Ces prototypes sont améliorés et modifiés jusqu'à l'obtention d'interfaces utilisateur conformes à des normes d'utilisabilité bien spécifiques (Konstan, 2011). L'évaluation précoce requiert un ou plusieurs évaluateurs expérimentés pour l'application de règles ergonomiques (ou d'heuristiques) sur un système donné (Salvendy et Turley, 2002). Parmi les approches existant pour l'évaluation précoce, nous pouvons citer celle de Tarby (Tarby, 2006; Tarby *et al.*, 2009). Cette dernière repose sur la programmation par aspect. Ce paradigme permet de greffer dès le début du cycle de développement d'un système interactif des traces orientées usage (Delannay, 2003) au niveau du noyau fonctionnel du système à concevoir. Ceci permet de faciliter la phase d'évaluation. Autrement dit, l'aspect évaluation des systèmes interactifs est pris en compte depuis la première phase de développement du système interactif. Comme la majorité des outils

existants s'appliquent à la phase finale du cycle de conception des systèmes interactifs, l'évaluation n'est pas prise en considération depuis les premières phases de conception. De plus, comme elle est délaissée à la fin, l'évaluateur se trouve souvent face à des contraintes d'ordre matériel et temporel pour procéder à l'évaluation. La motivation majeure de ce présent travail est de simplifier le processus d'évaluation et de prendre le maximum d'aspects lors de l'évaluation des interfaces utilisateur. De plus, le processus d'évaluation devrait être d'avantage automatisé afin de fournir des résultats plus fiables. Dans cette contribution, nous nous proposons d'automatiser le processus d'évaluation à l'évaluateur. De plus, nous nous proposons d'adopter une démarche d'évaluation précoce pour l'évaluation des interfaces utilisateur. Le processus d'évaluation proposé se base sur l'inspection de l'utilisabilité des *IU*et des tests utilisateur.

### 5.1.2 Concept

La présente contribution consiste à endoctriner des composants graphiques pour la composition de l'interface utilisateur avec des mécanismes permettant de supporter de telle évaluation. Ainsi l'évaluation s'effectue lors de la conception de l'interface utilisateur. Par conséquent, la conception de l'interface est couplée à la phase d'évaluation. En effet, dans cette contribution nous proposons des composants graphiques orientés évaluation. Ces composants proposés paraissent à la base similaires à ceux proposés habituellement par les *EDIs*[2] de type *WYSIWYG*[3] ; en effet, ils assurent les mêmes fonctionnalités et ainsi ils sont exposés sur la barre d'outils dans les environnements opérant avec le principe du « *Drag and Drop* », Figure 5.1. Aucune spécificité ne doit donc nécessairement être apparente à l'utilisateur de ces composants. Néanmoins, ils sont dotés de mécanismes supplémentaires permettant de supporter l'évaluation des interfaces utilisateur, *(Charfi *et al.*, 2012).

Autrement dit, parmi les fonctionnalités proposées à ces composants : à chaque fois où le concepteur ajoute un composant via le « *Drag and Drop* », ce dernier s'auto-inspecte par rapport à des règles ergonomiques préalablement définies. De plus, ce composant communique avec l'évaluateur grâce à une boite de message dédiée pour lui indiquer les éventuelles incohérences ergonomiques. Les composants dédiés à l'évaluation des interfaces utilisateur sont encapsulées sous formes de fichiers DLL afin de faciliter leur exploitation. Ils sont divisés suivant deux familles. La première inspecte l'utilisabilité de l'interface utilisateur. Elle se focalise sur la présentation statique de l'interface utilisateur. Chaque composant de cette famille, s'auto-inspecte par rapport à des règles

2. EDI est un programme regroupant plusieurs outils pour le développement d'application logicielle, de sites web, d'applications mobiles, etc.

3. WYSIWYG Ce sont les interfaces qui sont composées visuellement. Chaque fois que le concepteur ajoute ou modifie un composant graphique il voit immédiatement le résultat sur l'interface.

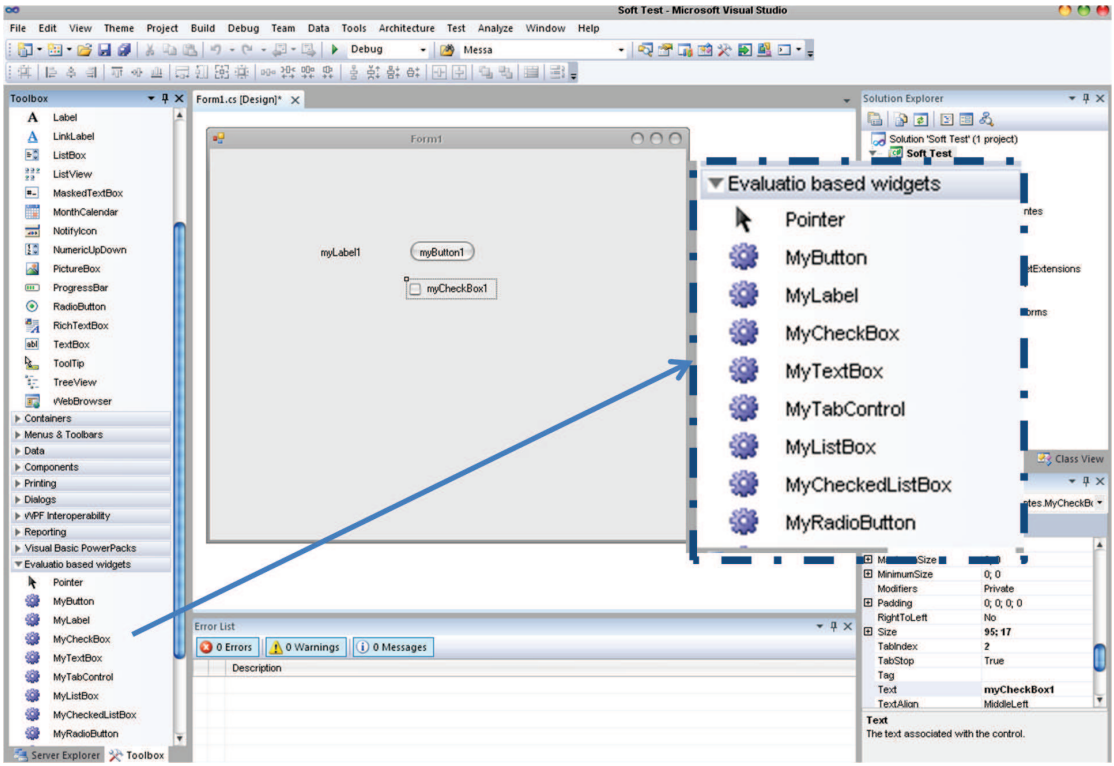



FIGURE 5.1: Exemple de composants orientés évaluation intégrés dans l'environnement « *MS Visual Studio 2010* »

ergonomiques stockées sous forme de fichiers XML externes à ces composants. La seconde famille est dédiée pour les tests utilisateur. Elle prend en considération l'interaction entre l'utilisateur et l'interface. Elle est basée sur un mouchard dédié à l'évaluation des *IU* pour la capture et l'analyse des séquences d'interactions. Le rôle des composants issus de cette famille est de communiquer les informations au mouchard (création, modification, utilisation, etc.). Un premier prototype de ces composants est développé pour être exploité sous l'environnement de développement « *Microsoft Visual Studio 2010*[4] ». Ils ont été implémentés suivant le langage de programmation C# (Framework 4.0). Les composants conçus sont : boîte combinée (combo-box), bouton radio, libellé, bouton, onglet, champ de texte, liste déroulante, case à cocher et liste déroulante de cases à cocher.

## 5.2 Composants dédiés pour l'inspection de l'utilisabilité

Comme mentionné lors du premier chapitre, les améliorations de l'interface en début du cycle de conception de l'interface utilisateur coûtent beaucoup moins qu'en fin du cycle (en termes de ressources matérielles et de temps). Les composants proposés s'intègrent dans cet esprit. Ils consistent à s'auto-inspecter au moment de leur création ou bien

4. http ://www.microsoft.com/france/visual-studio/default.aspx

leur modification (phase de conception de l'*IU*). Cette inspection s'effectue suivant des règles ergonomiques codées sous forme de fichiers XML. Une fois le composant est créé, ce dernier charge en mémoire l'ensemble des règles qui sont applicables à son type de composants graphiques. Ensuite, il s'inspecte suivant ces règles une par une. L'inspection s'effectue en comparant l'ensemble des valeurs de ses attributs à ceux spécifiés dans la règle. Une fois, il détecte une anomalie (par rapport aux règles inspectées), il charge l'erreur et les recommandations associées. Il avise par une notification le concepteur via cette erreur et les recommandations éventuelles pour corriger le problème détecté. Cette notification comporte ainsi les éventuelles incohérences ergonomiques par rapport aux règles spécifiées. Il expose les différents aspects du composant graphique qui ne sont pas conformes par rapport aux règles ergonomiques. Il contient aussi des suggestions pour remédier à l'incohérence du composant. En fin de l'inspection de la conformité par rapport aux règles ergonomiques, le composant génère un rapport englobant l'ensemble des erreurs de conception détectées dans l'interface graphique. Autrement dit ; les composants dédiés pour l'utilisabilité proposent deux types de rapports. Le premier rapport est spécifique à un composant graphique donné alors que le second est relatif à l'ensemble du système évalué. L'objectif des composants dédiés pour l'inspection de l'utilisabilité est de permettre lors de la conception une auto-évaluation par rapport à des règles ergonomiques spécifiées a priori par l'évaluateur.

### 5.2.1 Modélisation et exploitation des règles ergonomiques

Une règle ergonomique constitue, selon Vanderdonckt (Vanderdonckt, 1999), un principe de conception et/ou d'évaluation à observer en vue d'obtenir et/ou de garantir une interface Homme-Machine ergonomique. Généralement, une règle ergonomique est issue des travaux d'autres disciplines telles que le génie logiciel ou bien provient de l'observation ou l'étude des utilisateurs de systèmes interactifs. Elle est généralement exprimée dans un premier temps en langage naturel afin de guider le concepteur et/ou l'évaluateur pour l'obtention d'interfaces homme-machine utiles, accessibles et facilement utilisables. Avant de procéder à l'évaluation, les règles ergonomiques doivent être redéfinies en langage formel. Elles sont généralement exprimées dans les manuels ergonomiques en langage naturel d'où la difficulté de leur exploitation. Cette difficulté réside essentiellement dans leurs interprétations contextuelles (Keith, 2005). Dans la littérature, nous trouvons des langages dédiés à de telles définitions. Nous pouvons citer le « *Guideline Definition Langage* » (GDL) (Beirekdar *et al.*, 2002), le « *Guideline Abstraction Langage* » (GAL) (Leporini et Paterno, 2004) et le « *Unified Guideline Langage* » (UGL) (Arrue *et al.*, 2007b). L'utilisation de ces langages est assez compliquée et requiert des outils appropriés pour leur usage. De plus, ces langages sont conçus pour l'évaluation des interfaces

de type WUI. Dans cette contribution, nous nous sommes contentés d'une vision simpliste pour la modélisation des règles ergonomiques. Le choix du format XML pour la modélisation est dû à sa fiabilité et simplicité. Chaque règle ergonomique est définie par un fichier XML. La définition des règles se fait par l'évaluateur (ayant des connaissances en ergonomie logicielle) à travers un outil dédié à cet usage par l'évaluateur, Figure 5.2. Cet outil est baptisé « *The Ergonomic Guideline Manager* ».

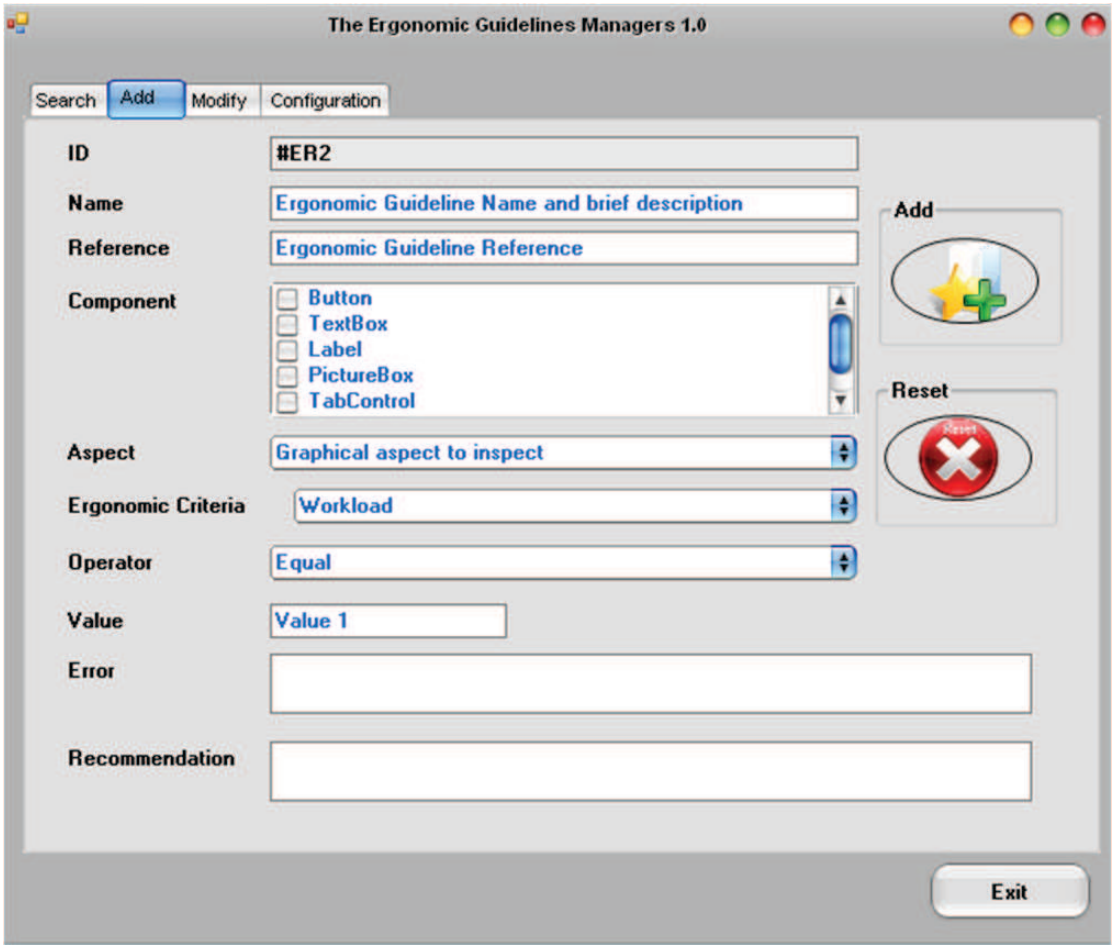



FIGURE 5.2: Capture écran de l'interface du « *The Ergonomic Guidelines Manager* »

L'outil, que nous proposons, définit les règles sous un format standard pour que les règles soient exploitables pour le processus d'évaluation. Cet outil, tel que l'illustre la Figure 5.2, permet de :

- Consulter les différentes règles ergonomiques existantes sous forme XML (*onglet Search* ). Cette recherche peut s'effectuer par l'identifiant de la règle, le nom ou bien la référence.
- Ajouter une nouvelle règle (*onglet Add* ) : l'identifiant de la règle est généré automatiquement par le système. L'évaluateur spécifie le nom et la référence bibliographique ainsi que l'ensemble des composants auxquels elle s'applique. Il dispose d'une liste des composants proposés. De plus, il exprime la règle à l'aide des opérateurs logiques

(égal, supérieur, inférieur, dans l'intervalle, dans l'ensemble, différent, différent de l'ensemble) et des attributs du composant (titre, fonte d'écriture, dimension, couleur, fond, etc.). Ensuite, il doit spécifier l'erreur générée lors de la non-conformité par rapport à cette règle et les suggestions de correction que le composant orienté évaluation peut fournir au concepteur.

– Modifier les règles existantes (*onglet Modify* ) : tous les champs peuvent être modifiés, à part l'identifiant de la règle.
– Configurer l'application (*onglet Configuration* ) en spécifiant le chemin de la sauvegarde des rapports d'évaluation ainsi que celui des fichiers XML.

Prenons l'exemple de la règle : « *La taille des caractères du texte principal doit être de douze (12) points. Les petites tailles de dix (10) points ou moins, doivent être réservées aux notes de bas de page, à certains éléments de navigation ou à d'autres informations comme la mention des droits d'auteur, de la date de mise à jour, etc.).* » (Millerand et Martial, 2001). La règle est définie en XML comme l'illustre le code ci-dessous.

```
<Style>
 <EG_ID>#ER2</EG_ID>
 <EG_Name>Fontsize</EG_Name>
 <EG_Controls>Button;TextBox;Label;ComboBox;</EG_Controls>
 <EG_Aspect>Font.Size</EG_Aspect>
 <EG_Operator>Between (two value)</EG_Operator>
 <EG_Value1>10</EG_Value1>
 <EG_Value2>12</EG_Value2>
 <EG_Error>Font size</EG_Error>
 <EG_Recommendation>The main text font size must be twelve (12) points.
 The small size have to be of ten (10) points or less; it should be reserved for
 footnotes page, certain navigation elements or other information such as
 the copyright mention, update date etc.</EG_Recommendation>
 <EG_Density>30</EG_Density>
</Style>
```

La modélisation avec l'outil proposé consiste tout d'abord dans le choix des règles à prendre en considération pour la phase de conception/évaluation. L'évaluateur doit spécifier l'aspect graphique sur lequel porte la règle (police, taille, dimension, couleur, etc.). Puis il détermine l'ensemble des composants graphiques auxquelles s'applique la règle. Ensuite, il exprime la règle à l'aide d'opérateurs arithmétiques (supérieur, inférieur, différent, etc.). Enfin, l'évaluateur associe à la règle l'erreur provoquée ainsi que les suggestions d'amélioration. La règle est enregistrée ensuite sous forme XML dans un répertoire spécifique aux composants, Figure 5.3.

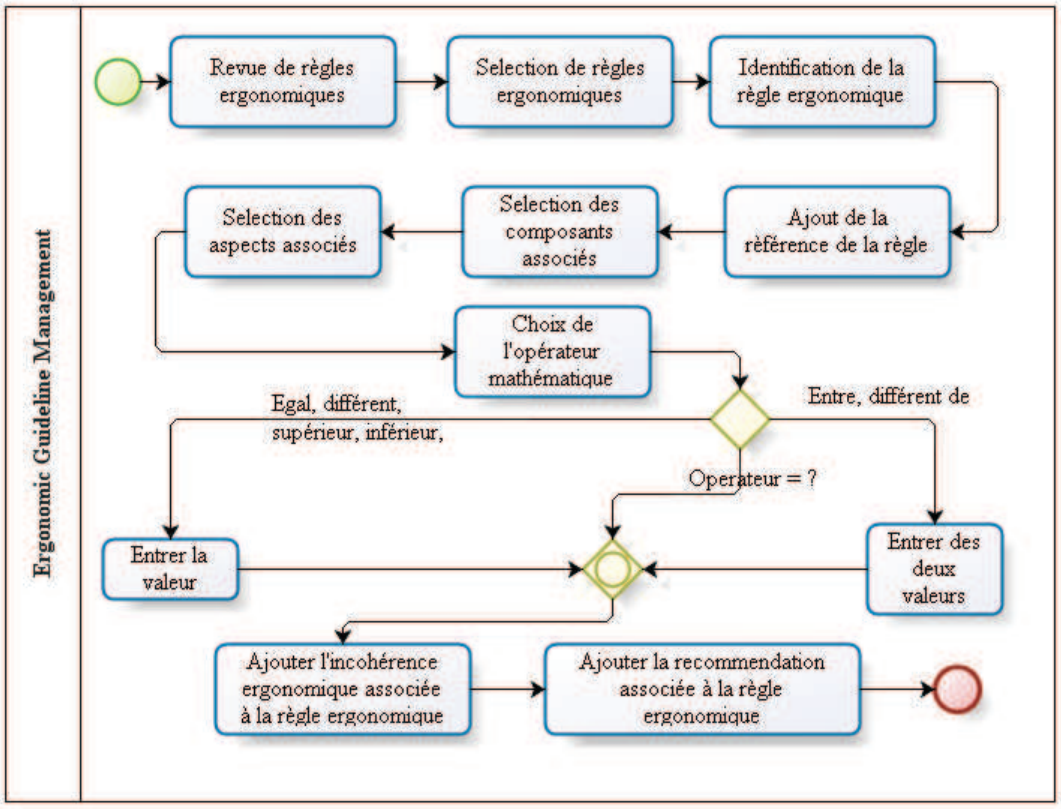



FIGURE 5.3: Processus de modélisation des règles ergonomiques sous forme de fichier XML.

### 5.2.2 Exploitation des règles ergonomiques au sein des composants pour l'évaluation

Lors de son « *auto-évaluation* », le composant parcourt l'ensemble des règles ergonomiques spécifiées par l'évaluateur. Il duplique en mémoire les règles dont son type figure dans la balise « *EG_controls* » du fichier XML de la règle. Une fois le parcours terminé il inspecte sa conformité par rapport aux règles une par une. Pour chaque règle, le composant identifie en premier lieu l'opérateur logique sélectionné (inférieur, supérieur, égal, etc.). A chaque opérateur correspond une procédure dans laquelle en argument, le composant intègre la (les) valeur(s) recommandée(s) ainsi que les attributs. A chaque fois que le composant inspecte une règle et se trouve en non-conformité avec celle-ci, les chaines de caractères « *recomm* » et « *error* » sont alimentées de l'erreur de conception détectée ainsi que de la suggestion associée. En fin de protocole d'inspection, le composant affiche ces deux chaines de caractères à l'évaluateur pour l'aviser des éventuelles incohérences ergonomiques sur l'interface en cours de conception. Le code ci-dessous illustre le processus de l'auto-évaluation au sein des composants graphiques proposés. Tout d'abord, le composant s'initialise sur l'interface graphique en exploitant le constructeur hérité.

Ensuite, il charge l'ensemble des règles ergonomiques associées à son type dans une file. Puis, il analyse l'ensemble des règles suivant le type d'opérateurs logiques et algorithmiques utilisés (inférieur, supérieur, égal, différent, etc.) A chaque opérateur est associée une méthode dont le composant lui passe en argument les valeurs de ses attributs et la règle ergonomique. En fin de parcours de la file de règle, le composant orienté évaluation notifie le concepteur par un message des éventuelles violations des règles ergonomiques et enregistre les incohérences sous forme de fichier PDF (le rapport de l'évaluation).

```
General evaluation based controls algorithm
Input:  Ergonomic guidelines XML files
        Provided design environment controls
Output: Conformity notification
        Evaluation report
Begin
 Apply inherited constructor
 Initialize the control on the graphic interface
 Read XML files
 Load XML file into an array list
 Array-list parse
 Load element (counter)
 While(counter<=arraylist.count)
 Loop
 Switch(operator)
 Case "Inferior": inferior(arraylist(i)); break;
 Case "Superior": superior(arraylist(i)); break;
 Case "Equal": equal(arraylist(i)); break;
 ...
 ...
 Counter++
 End loop
 Notification(errors, recommendation).show();
 Save(errors, recommendation, report);
End
```

## 5.3 Composants dédiés pour les tests utilisateur

Afin d'assurer des tests utilisateur de l'interface utilisateur, les composants proposés exploitent un mouchard électronique. Les composants dédiés évaluation et le mouchard sont structurés autour d'une architecture client-serveur afin d'assurer un dialogue entre eux. Dans cette architecture le mouchard est le serveur et chaque composant est un client. La partie serveur et les clients sont liés via le réseau TCP/IP. L'évaluation des interfaces utilisateur à travers ces composants et le mouchard vise à détecter les problèmes relatifs à l'utilisation d'un système interactif, Figure 5.4. A chaque action utilisateur, le composant avise le mouchard par l'ensemble des informations associées : l'identifiant du composant, le type du composant, la nature de l'action (clic souris, clic droit, etc.),

le temps d'exécution, l'interface à laquelle il appartient, le contenu du composant et l'adresse IP de la machine cliente.

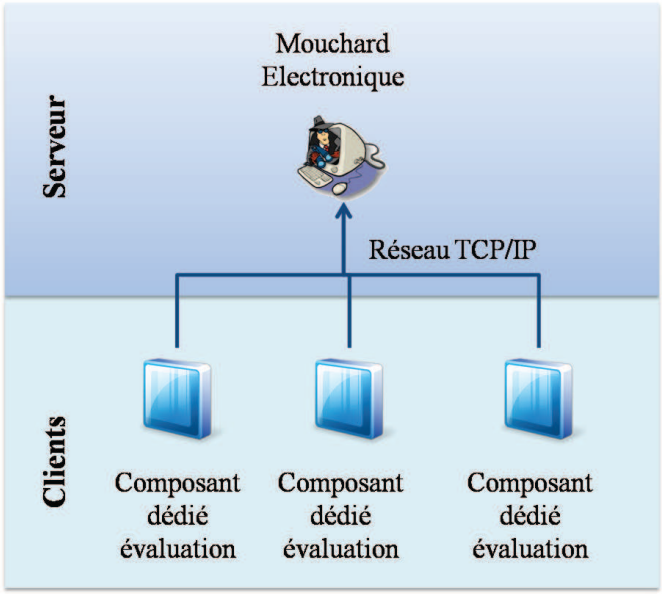



FIGURE 5.4: Architecture logicielle logique du mouchard électronique et des composants orientés évaluation

### 5.3.1 Mouchard Electronique

Le mouchard électronique, Figure 5.6, considère l'évaluation suivant le principe énoncé par Senach (Senach, 1990). L'évaluation s'effectue en comparant deux modèles. Le premier est un modèle de référence spécifié par l'évaluateur et le second est le modèle observé qui concerne les actions exécutées par l'utilisateur. Ainsi l'évaluation se base sur des données empiriques. Le mouchard électronique est structuré autour de quatre modules, Figure 5.5.

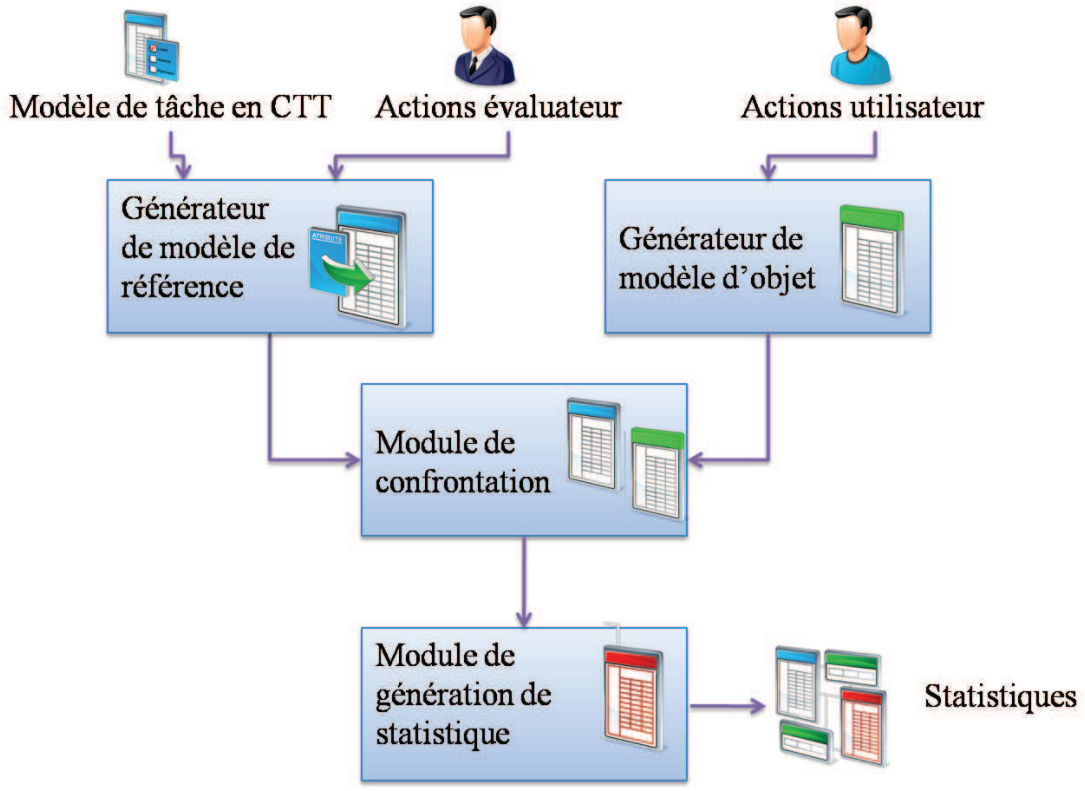



FIGURE 5.5: Architecture fonctionnelle du mouchard électronique

#### 5.3.1.1 Le générateur de modèles de référence

Le premier module permet de générer le modèle de référence. Il se base sur la notation CTT pour la modélisation de ce modèle. Les tâches utilisateurs sont formalisées suivant l'environnement CTTE suivant la notation CTT (Paterno *et al.*, 1997; Paterno, 2000). Ensuite, l'évaluateur à travers ce module, associe les différentes actions élémentaires aux tâches utilisateurs, Figure 5.6. Ces actions élémentaires sont identifiées à partir des interactions entre l'évaluateur et l'interface utilisateur. Le mouchard les détecte et les enregistre en mémoire. Autrement dit, la modélisation de l'hiérarchie et des relations entre les tâches s'effectue grâce à l'outil CTTE. Par contre, l'attribution des actions utilisateur aux tâches et sous-tâches s'effectue avec le présent module.

La notation CTT (Paterno *et al.*, 1997; Paterno, 2000) a été choisie pour la modélisation des tâches du fait que celle-ci met l'accent sur les activités de l'utilisateur et elle propose de nombreux opérateurs qui prennent en considération l'aspect temporel lors de la modélisation. L'utilisateur dispose ainsi de certains opérateurs tel que l'activation (>>), l'entrelacement (|||), synchronisation (|[ ]|), l'interruption (|>), la désactivation (>), etc. La modélisation des tâches en CTT s'élabore généralement suivant trois phases. La première concerne la décomposition hiérarchique des tâches en sous-tâches sous forme

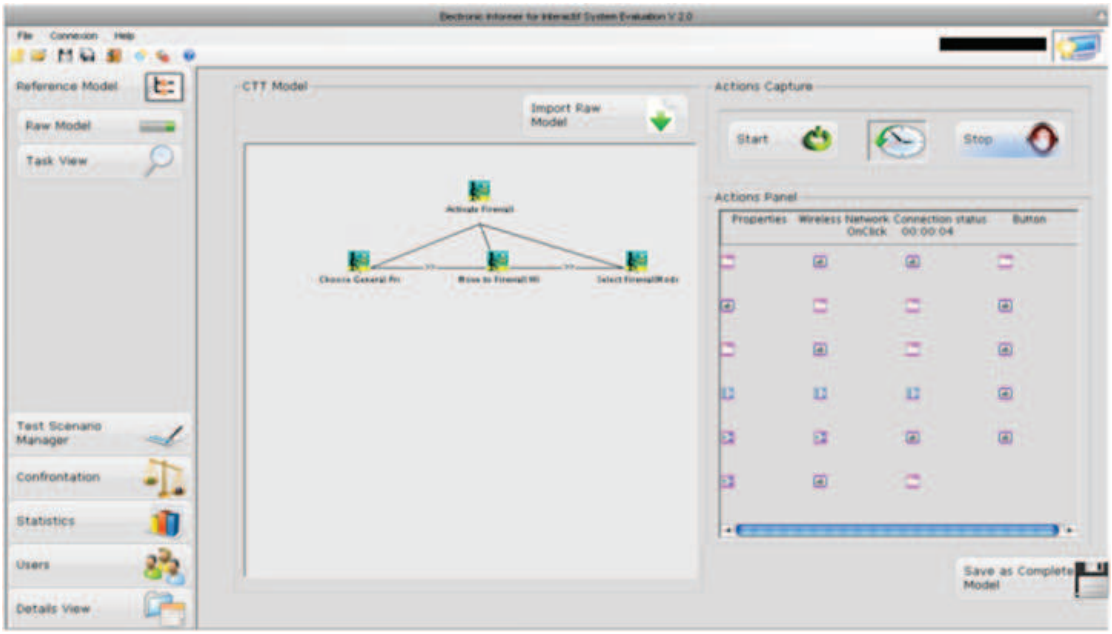



FIGURE 5.6: Interface du mouchard électronique dédié à l'évaluation des IU

d'arbre. La seconde consiste à identifier les liens temporels entre ces tâches et sous-tâches. Enfin, la troisième phase se rapporte à identifier les objets et les actions associés à chaque sous-tâche (Navarre, 2001; Winckler *et al.*, 2004).

#### 5.3.1.2 Le générateur de modèles de la tâche observée

Ce module est chargé de détecter et d'enregistrer les actions de l'utilisateur dans le but de constituer le modèle de la tâche observée. La capture s'effectue en réceptionnant les différentes informations citées ci-dessus depuis les composants graphiques. Les informations sont identifiées à partir des trames de données envoyées par les composants lors de l'interaction avec l'utilisateur. Ces actions élémentaires sont enregistrées sous forme de fichiers XML. La capture des données ne s'effectue pas pour un ensemble de tâches à la fois, elle s'effectue tâche par tâche dans le but de simplifier le processus d'exploitation de ces données pour l'évaluation des interfaces utilisateur. Par contre, ce module permet de capturer des différentes séquences d'interaction issues des interactions de différents utilisateurs en même temps. Il exploite le champ adresse IP afin de délimiter les actions de chaque utilisateur. L'interaction simultanée de plusieurs utilisateurs lors du test permet d'élaborer plus rapidement la phase de test utilisateur et notamment dans le cas où l'évaluateur fait appel à plusieurs utilisateurs (de l'ordre de quelques dizaines et plus).

#### 5.3.1.3 Le module de confrontation

Ce module permet d'élaborer une confrontation entre les deux modèles issues des deux premiers modules. Cette comparaison vise notamment par la détection d'actions répétitives, erronées et inutiles. La comparaison s'effectue en exploitant les automates à états finis (Anderson et Head, 2006). Le choix des AEF est dû au fait que d'une part ils présentent une simplicité au niveau de la mise en place et de l'exploitation. D'autre part, les AEF permettent de modéliser l'exécution d'une tâche quand il y a plusieurs façons de procéder. A chaque tâche est associé un automate à états finis. Le modèle de référence est ainsi défini par un quintuple ($\sum$, S, $s_0$, d, F) :

- $\sum$ : l'alphabet d'entrée. Elle contient l'ensemble des actions élémentaires qui peuvent être exécutées par l'utilisateur. Cet ensemble est constitué par des éléments qui sont sous le forme de couple (composant graphique, action) ;
- S : l'ensemble des états du système. Chaque état correspond à une sous-tâche qui peut être exécutée avec le système. Notons qu'une tâche est composée par une ou plusieurs actions élémentaires. Chaque tâche est modélisée par un symbole ;
- $s_0$ : L'ensemble des états initiaux. card($s_0$) $> 0$ ;
- d : est une fonction de transition entre les différents états. d : S x P - > S et,
- F : est l'ensemble des états finaux. Tout comme $s_0$, card(F) $> 0$.

```
L'alphabet intitulée SAI_EVALUATION est définie par (A, S, s0, d, F):
A = {ButtonClick1, ButtonClick2, ButtonClick 39, TextBox1, TextBox2,
TextBox16, etc.},
S = {Launch SAI, MessageBus61, MessageBus11, MessageBus17, MessageBus3,
MessageRailway1, MessageRailway2, MessageRailway3, MessageRailway9, CloseSAI, etc.},
s0 = {LaunchSAI},
d: {(LaunchSAI, ButtonClick6)-> MessageBus61,
(MessageBox6, ButtonClick6)-> MessageBox6,
(LaunchSAI, ButtonClick8) -> MessageBux13,etc.)
F = {CloseSAI}.
```

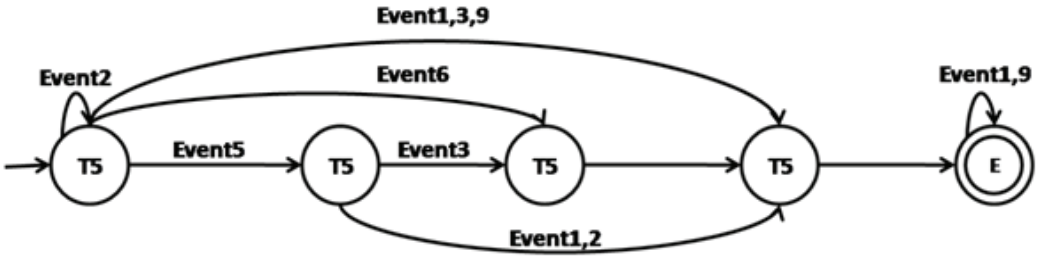



FIGURE 5.7: Exemple d'automate à états finis exploité pour l'évaluation des systèmes interactifs

Cet automate est élaboré en convertissant le CTT (avec les actions élémentaires associées). Le modèle observé quant à lui est représenté par un mot (au sens des AEFs)

constitué par les différentes actions élémentaires exécutées par l'utilisateur. La confrontation consiste ainsi à vérifier si ce mot est un mot de l'automate associé au modèle de référence. Dans le cas où la liste des actions élémentaires n'est pas un mot de l'automate spécifié, ce module identifie les actions manquantes ou encore inutiles pour qu'il soit un mot de ce grammaire. Ces actions permettent d'établir ce que l'utilisateur n'a pas fait par rapport au modèle de référence et ses actions inutiles et par conséquent identifier les problèmes de l'utilisation de l'interface utilisateur. Notons qu'à chaque tâche exécutée par un utilisateur correspond un mot.

#### 5.3.1.4 Le module de génération de statistiques

Le quatrième module permet de raffiner les résultats de comparaisons détectées par le troisième module dans le but de les présenter à l'évaluateur. Il permet de générer des statistiques afin d'assister l'évaluateur lors de l'évaluation de l'interface utilisateur. Ces statistiques portent essentiellement sur le pourcentage de l'exécution d'une tâche, le taux d'accomplissement des sous-tâches, et la comparaison entre le modèle de référence et le modèle observé suivant une représentation graphique simplifiée. Cette dernière consiste à générer l'arbre de la tâche décomposée en sous-tâches. Chaque sous-tâche est colorée pour indiquer l'état d'exécution de la tâche, Figure 5.8 :

- Vert : tâche exécutée avec succès,
- Jaune : tâche incomplète ;
- Rose : tâche exécutée avec succès mais avec un retard au niveau du temps d'exécution et,
- Rouge : échec de l'exécution de la tâche.

Ce module permet aussi de renseigner l'évaluateur avec certaines statistiques par rapport à une tâche donnée :

- Le nombre d'utilisateur qui ont exécuté cette tâche ;
- Le nombre d'utilisateurs qui ont réussi à exécuter cette tâche en toute sa totalité ;
- Le nombre d'utilisateurs qui ont réussi à exécuter partiellement cette tâche avec le pourcentage de réussite de l'exécution et,
- Le taux de succès lors de l'exécution de la tâche par utilisateur.

De plus, il permet de consulter les différentes actions exécutées par chaque utilisateur.

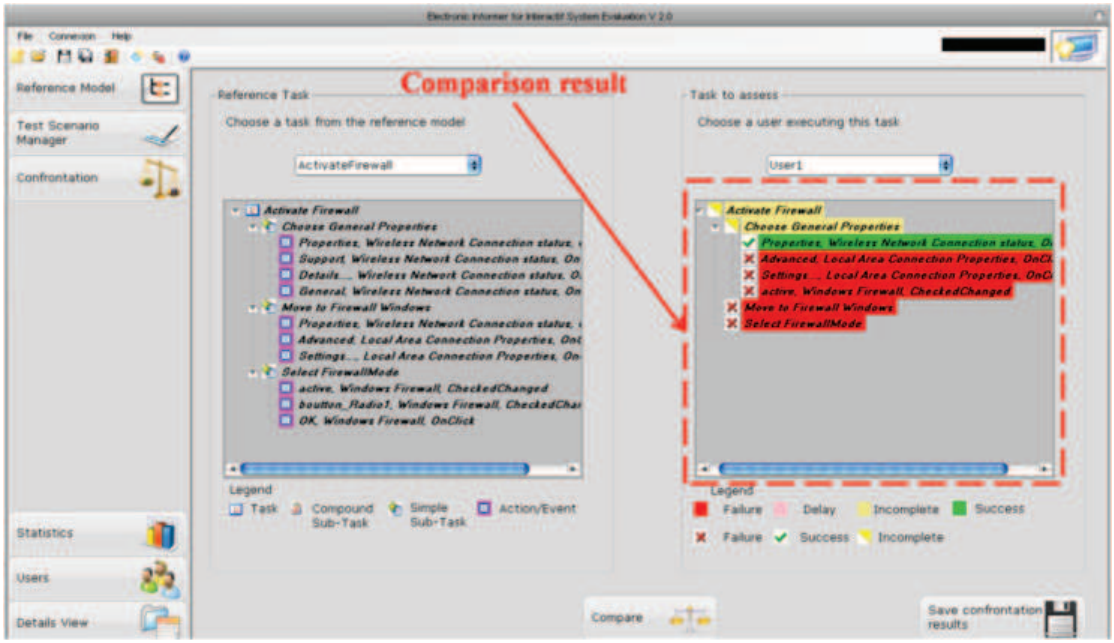



FIGURE 5.8: Exemple de résultats proposés par le mouchard électronique dédié à l'évaluation des IU

## 5.4 Processus d'évaluation adopté

Nous avons vu que les composants orientés évaluation s'étendent essentiellement sur deux familles dédiée respectivement à l'inspection de l'utilisabilité et des tests utilisateur. Nous allons dans cette section détailler le processus d'évaluation des interfaces utilisateur pour chaque famille.

### 5.4.1 Processus de l'évaluation de l'utilisabilité à travers les composants orientés évaluation

L'évaluation de l'utilisabilité des interfaces utilisateur à travers les composants orientés évaluation s'opère sur deux phases. La première phase est celle du pré-évaluation/conception et la seconde est une phase d'évaluation/conception.

#### 5.4.1.1 Phase de pré-évaluation/conception

Cette phase consiste à injecter dans le code source des composants graphiques des mécanismes spécifiques. Ces mécanismes concernent un algorithme inspectant la conformité du composant lors de sa création par rapport aux règles ergonomiques. Dans cette phase, l'évaluateur spécifie les règles ergonomiques à inspecter la conformité par les composants.

#### 5.4.1.2 Phase d'évaluation/conception

Une fois le composant graphique est ajouté à l'interface utilisateur en cours de conception, il exécute ces trois actions :

- Chargement des règles ergonomiques associées à son type (bouton, case à cocher, liste de choix, etc.). Le choix s'effectue en parcourant les différentes règles et à copier en mémoire celles dont la balise control contient son type.
- Extraction des valeurs des attributs graphiques : le composant extrait les valeurs de ses attributs et les stocke dans une mémoire.
- Comparaison entre recommandations ergonomiques et valeurs des attributs du composant : le composant parcourt la file des règles ergonomiques et vérifie sa conformité par rapport aux règles. Chaque fois qu'une anomalie est détectée, le composant charge en mémoire structure cette violation par l'ajout de l'erreur et les recommandations associées.

A la fin du parcours de la liste des règles ergonomiques, le composant notifie à l'évaluateur et au concepteur l'ensemble des incohérences détectées. Ceci permet de dégager d'une façon automatique les problèmes d'utilisabilité. A noter que cette procédure est déclenchée chaque fois qu'un composant graphique est modifié.

### 5.4.2 Processus de l'évaluation de l'utilité à travers les composants orientés évaluation

Le processus du test utilisateur dans les interfaces utilisateur est décrit dans la figure ci-dessous. Ce processus s'étend sur trois phases. La première consiste à préparer les données pour le processus de l'évaluation. La seconde est l'évaluation et la troisième concerne la phase de post-évaluation.

#### 5.4.2.1 Phase de pré-évaluation

Cette phase correspond à la mise en place des composants dédiés évaluation par un (ou plusieurs) développeur(s) ; les encapsuler sous forme de fichiers DLL et les intégrer au *EDI*. Ensuite, le concepteur se charge de composer graphiquement l'interface utilisateur du système interactif à évaluer à partir des composants proposés sur la barre d'outils, Figure 5.1. Entre temps, l'évaluateur spécifie le modèle de tâche à l'aide de l'environnement CTTE avec la notation CTT. Rappelons qu'à chaque tâche à effectuer lors de test utilisateur, correspond un modèle. Ces modèles sont enregistrés sous forme de fichiers XML.

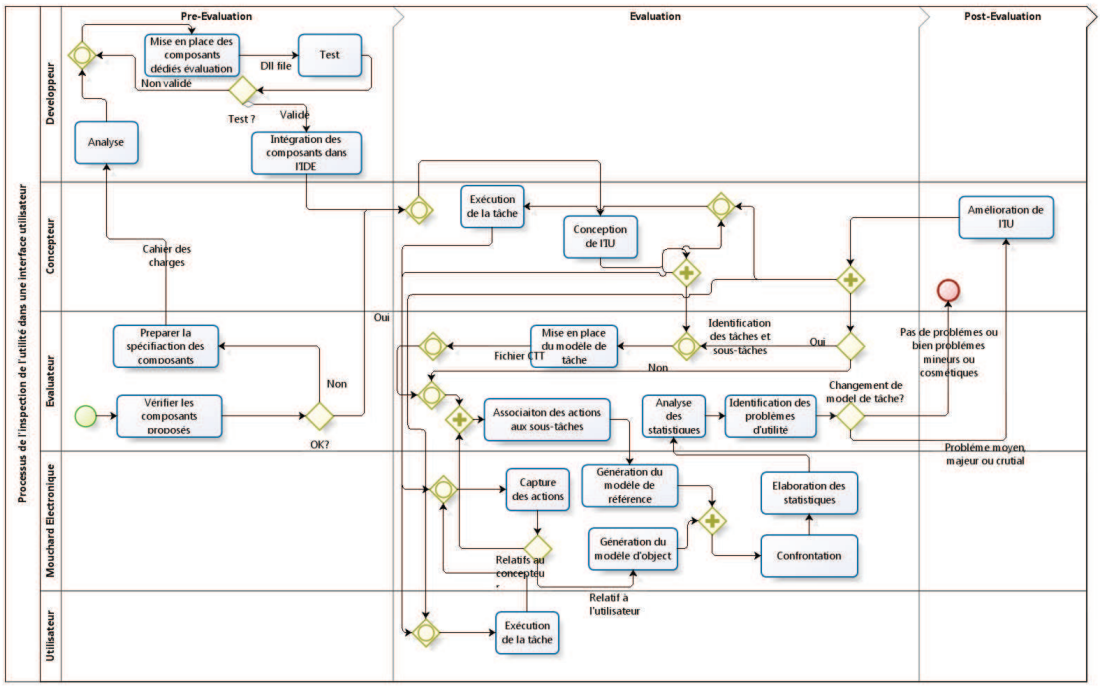



FIGURE 5.9: Processus de l'évaluation à travers les composants dédiés pour les tests utilisateur

### 5.4.2.2 Phase d'évaluation

Une fois, la modélisation des tâches est élaborée, le concepteur exécute les différents tâches une par une. Le mouchard détecte ainsi l'exécution de cette tâche et l'enregistre comme étant un modèle de référence grâce à l'association du modèle spécifié en CTT et la liste des actions élémentaires exécutées par le concepteur. Cette association s'effectue par l'évaluateur grâce au module générateur de modèles de référence. Une fois, le modèle de référence est élaboré, l'évaluateur peut exécuter les tests utilisateurs. Ces tests concernent l'exécution de la tâche par plusieurs utilisateurs dans le but de comparer leurs actions à celle du concepteur. Chaque fois où l'utilisateur exécute une action sur l'interface, le composant associé avise le mouchard par un message. Le mouchard est chargé de capturer toutes les données concernant l'interaction avec l'utilisateur. A la fin de la session d'exécution, le mouchard électronique dispose des différentes séquences d'interaction des différents utilisateurs. Ces séquences sont sauvegardées dans des fichiers XML. Leur gestion est effectuée par le module de générateur de modèles d'objet. Ces fichiers XML sont transférés par la suite au troisième module qui compare entre les différents modèles à l'aide des AEFs. Cette confrontation relève les différents problèmes d'utilisation de l'interface. Ces problèmes sont communiqués à l'évaluateur via une interface graphique dédié. L'évaluateur recense ainsi l'ensemble de problèmes détectés dans l'interface utilisateur afin de fournir un rapport d'évaluation contenant les différents problèmes et les suggestions d'amélioration. L'interface est par la suite rectifiée suite à

ce rapport par le développeur. Le processus d'évaluation est itératif dans le processus d'évaluation proposé.

#### 5.4.2.3 Phase de post-évaluation

Cette phase consiste à apporter des améliorations sur l'interface suite aux problèmes détectées dans l'interface utilisateur ainsi que les suggestions et les remarques de l'évaluateur. Le processus peut être itératif en procédant de nouveau à l'évaluation de la nouvelle interface.

## 5.5 Synthèse et discussion

L'évaluation des interfaces utilisateur est une tâche primordiale pour l'obtention d'interfaces utiles et utilisables (Greenberg et Buxton, 2008). Lors de ce chapitre, nous avons présenté une contribution pour l'évaluation des interfaces utilisateur. Cette contribution s'intègre dans le cadre de l'évaluation précoce. Cette approche se base sur des composants graphiques pour la composition d'interface graphique sous les *EDI* de type *WYSIWIG*. Ces composants ont la spécificité de supporter l'évaluation des interfaces graphiques. Ceci permet de coupler entre la phase d'évaluation et la phase de conception. Cette spécificité est réalisée suite à l'endoctrinement des composants graphiques proposés par des mécanismes permettant d'évaluer les *IUs* ou de supporter leur évaluation. A noter que ces composants ne présentent pas d'aspects apparents concernant le processus d'évaluation. Ils paraissent aux concepteurs de l'interface comme étant des composants classiques pour la conception d'interfaces graphiques via le concept du « *Drag and Drop* ».

Les composants orientés évaluation s'articulent essentiellement autour de deux familles. La première concerne l'ensemble des composants qui s'auto-inspectent eux-mêmes leur conformité par rapport à des règles ergonomiques. Une fois, qu'ils détectent une incohérence avec une ou plusieurs règles, ils avisent le concepteur afin de remédier à cette erreur de conception. Cette collection de composants vise à détecter les problèmes d'utilisabilité dans une interface utilisateur, leurs fréquences, leurs impacts et leurs persistances. Cette collection présente un ensemble d'avantages par rapport aux solutions existantes. Tout d'abord, elle propose un processus d'évaluation totalement automatique pendant les trois phases de l'évaluation (capture, analyse et critique au sens de (Ivory et Hearst, 2001)). L'intervention de l'opérateur humain n'a lieu que pour modéliser et spécifier les règles ergonomiques à prendre en considération lors de la validation de la conformité de l'interface par rapport à ces données. Ensuite, ces composants proposent une assistance

pour le concepteur afin d'élaborer des interfaces de bonne qualité ergonomique et par suite utiles et utilisables.

La seconde collection opère avec un mouchard électronique dédié à l'évaluation des interfaces utilisateur. La tâche des composants dans cette collection se restreint à informer le mouchard des opérations qu'ils subissent (création, clic sur un bouton, saisie de texte, etc.). Le mouchard par la suite permet de constituer le modèle de référence et le modèle observé dans le but de les comparer afin de dégager les problèmes d'utilisation. La comparaison se base sur les AEFs. Cette collection vise à identifier les problèmes que peut connaitre l'utilisateur en interagissant avec l'interface graphique. L'avantage de ce mouchard est le fait que contrairement à la majorité des mouchards électroniques dédiés à l'évaluation des *IU*, analyse les données capturées afin de présenter une synthèse de ces données. Il ne propose pas d'immenses quantités d'information qui requiert du temps et de l'effort pour les traiter *(Charfi *et al.*, 2011a). En effet, l'un des inconvénients de l'utilisation des mouchards électroniques concerne le traitement et l'analyse de la grande quantité d'informations capturées. De plus, le mouchard proposé couvre un large spectre des données capturées tel est le cas des actions élémentaires et le temps d'exécution. D'ailleurs, il prend en considération la décomposition hiérarchique des tâches en sous-tâches et en actions élémentaires via l'adoption de la notation CTT pour modéliser les différentes tâches. En outre, ce mouchard ne requiert pas la modification du code source du système à évaluer pour procéder à l'évaluation tel que les mouchards EMA (Balbo, 1996) ou bien EISEval (Tran *et al.*, 2008).

L'avantage de ces propositions réside essentiellement dans le fait de supporter le processus d'évaluation depuis les premières phases. Nous avons vu dans le deuxième et le troisième chapitre, que la majorité des outils existants ne s'applique que pour un système réalisé ou encore à des prototypes avancés d'un système interactif. Ainsi le processus d'évaluation s'avère plus difficile et délicat à mettre en place. De plus, les composants proposés montrent une facilité d'utilisation pour leur exploitation en but d'évaluer les *IUs*. En effet, il doit être facile de concevoir des maquettes de l'interface à réaliser avec les composants proposés. Le concepteur de l'interface n'a pas à vérifier manuellement la conformité des composants par rapport aux règles ergonomiques.

Notons que les composants proposés dans ce chapitre sont destinés pour l'évaluation des interfaces de types WIMP. Nous avons développé d'autres composants destinés pour l'évaluation des interfaces Mobile et WUI. Au delà de la nature de l'interface évaluée, la différence réside aussi au niveau de l'architecture fonctionnelle adoptée. Les composants pour les interfaces WIMP imbriquent des mécanismes d'évaluation au sein de ces composants. Par contre, avec les composants de type WUI et Mobile, nous avons réalisé une

application s'exécutant en arrière plan afin de communiquer avec le mouchard. Cette application est intitulée « *Médiateur* ». En effet, elle sert de médiateur entre :

– l'utilisateur et le composant pour communiquer les différentes erreurs de conception liées au composant dans le cas de l'inspection de l'utilisabilité et,
– le composant et le mouchard dans le cas du test utilisateur.

La liaison entre ce médiateur avec le composant ainsi que le mouchard s'effectue en local via l'utilisation des sockets. Ce Médiateur est chargé de détecter les différentes informations relatives aux attributs des composants graphiques ainsi que les règles ergonomiques associées, Figure 5.10.

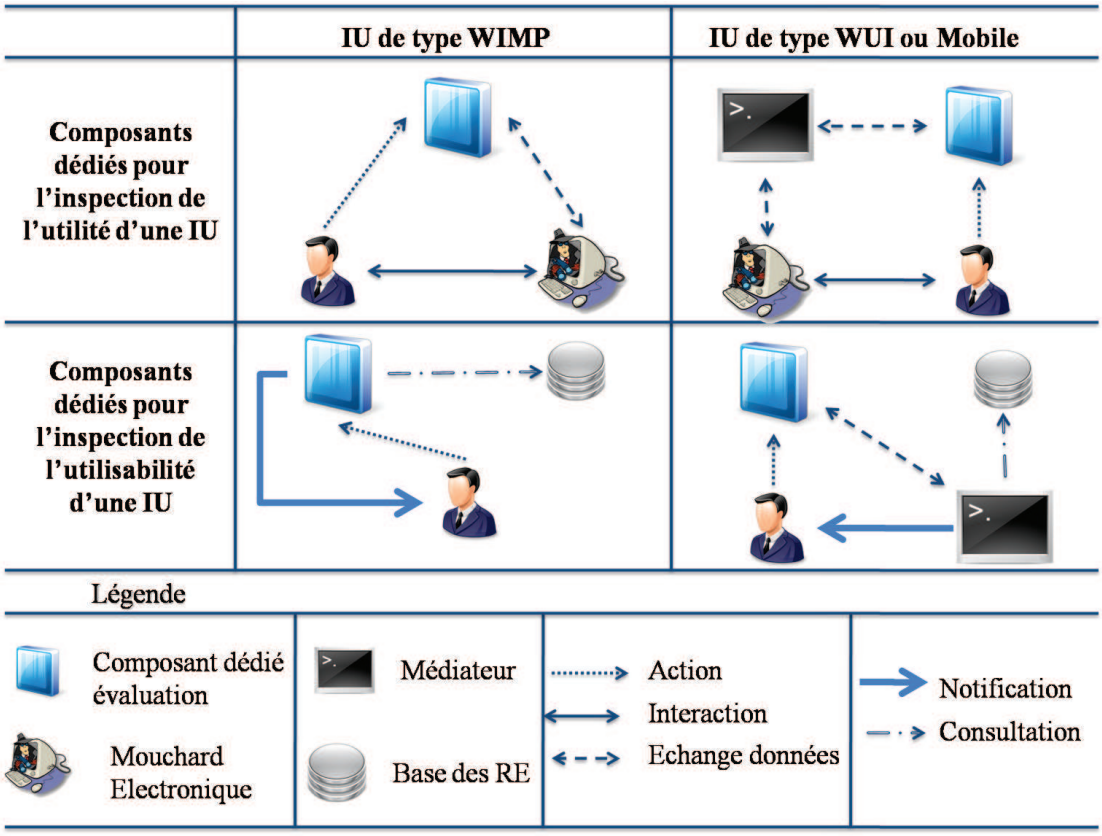



FIGURE 5.10: Architecture fonctionnelle des composants orientés évaluation

Parmi les limites de la contribution proposée, nous pouvons citer le fait que ni la première famille de composant et ni la seconde ne permettent d'indiquer la gravité des problèmes identifiés dans les interfaces utilisateur. Ensuite, le rapport présenté par le mouchard nécessite davantage de travail. En effet, ce rapport se restreint à montrer l'état d'achèvement des différentes tâches par l'utilisateur (réussite, échec, retard et incomplétude). Le résultat de comparaison entre le modèle observé et le modèle de référence peut être mieux élaboré dans le but de mieux assister l'évaluateur à identifier les problèmes d'utilisation dans une interface graphique.

# Troisième partie

# Mise en oeuvre

Chapitre 6

# Evaluation expérimentale

> « *Avant de vouloir qu'un logiciel soit réutilisable, il faudrait d'abord qu'il ait été utilisable* »
>
> Ralph Johnsono

## Introduction

Au cours des chapitres précédents, nous avons dans un premier lieu dressé un état de l'art des outils existants pour l'évaluation des interfaces utilisateur ainsi que de leurs limites (Chapitres 1, 2 et 3). Dans un second lieu, nous avons présenté deux contributions pour l'évaluation des *IHM*. Lors de ce chapitre, nous nous proposons de décrire deux évaluations pratiques mettant en application les deux contributions proposées dans les chapitres 4 et 5. Notons que l'objectif de ces deux évaluations n'est pas d'évaluer les systèmes interactifs en soi mais plutôt d'appliquer et de tester les deux contributions proposées dans ce présent mémoire. La première évaluation consiste à tester et à valider le framework *RITA*. La seconde évaluation vise à tester les composants orientés évaluation dans le but de délimiter les fonctionnalités et l'efficacité des composants proposés. Ces deux évaluations s'effectuent dans le cadre d'un projet intitulé « *CISIT-ISART* ». Elles consistent à évaluer un système dédié pour la supervision d'un réseau de transport afin de dégager les aspects engendrant des problèmes d'utilisation de ce système. Notons que ces deux évaluations se déroulent en laboratoire. Lors des deux évaluations, nous présenterons le processus d'évaluation, le dispositif expérimental associé et les résultats obtenus. Ce chapitre sera conclu avec une discussion portant sur les deux contributions et de leurs limites.

## 6.1 Première évaluation : évaluation du framework RITA

Cette section vise à présenter une évaluation expérimentale de la contribution présentée lors du quatrième chapitre. Rappelons qu'elle s'articule autour d'un framework pour l'évaluation des Interfaces Utilisateurs. Ce framework a l'avantage d'être essentiellement configurable, générique et multi-techniques. Le test de ce framework porte sur l'évaluation d'un système dédié pour la supervision du réseau de transport : Le SAI. Ce système est décrit dans la sous-section suivante.

### 6.1.1 Présentation du système à évaluer : SAI

Dans le cadre du projet « *SART*[1] », un système a été proposé dans le but d'assister les régulateurs en salle de contrôle pour la gestion et la supervision d'un réseau de transport urbain. Il vise à ce que les régulateurs accomplissent leurs tâches d'une façon optimale lors d'un fonctionnement normal ou encore anormal du réseau de transport (Bonte *et al.*, 2007). Ainsi, le projet SART vise à minimiser le temps d'attente des voyageurs et à leur assurer la continuité des déplacements dans les réseaux multimodaux (Bonte *et al.*, 2007; Tran, 2009). Le système proposé dans le cadre du projet SART s'articule autour de trois composantes 6.1.

1. SART est un projet coopératif du Groupement Régional Nord-Pas-de-Calais pour la Recherche dans les Transports. Il vise à aider les voyageurs en améliorant la qualité des correspondances dans les réseaux de transport urbains en les tenant informés de l'état du trafic. Il fait intervenir un partenaire industriel (Tranvilles) et des composantes de recherche (LAMIH, LAGIS et l'IFSTAR) (Bonte *et al.*, 2007).

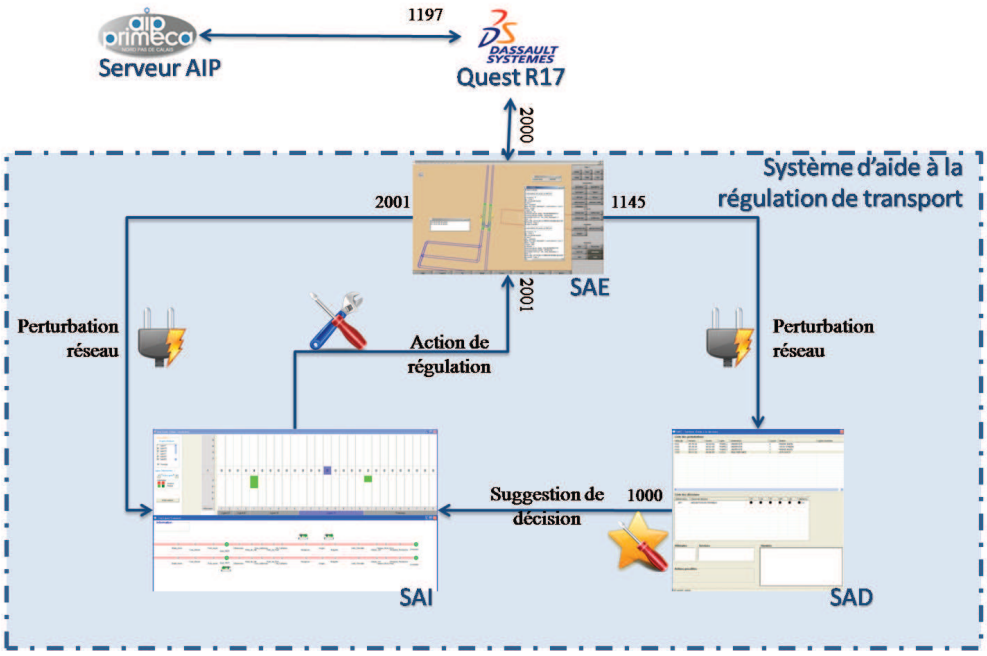



FIGURE 6.1: Architecture du système de supervision proposé dans le cadre du projet SART

– Le SAI : ce système est chargé de présenter les différentes informations concernant le réseau de transport aux voyageurs via les régulateurs, Figure 6.2. Il a essentiellement comme objectif d'informer entre autre les superviseurs du positionnement des différents véhicules dans le réseau de transport. De plus, il permet aussi au régulateur de communiquer avec les conducteurs des différents véhicules via l'envoi de messages, Figure 6.2.

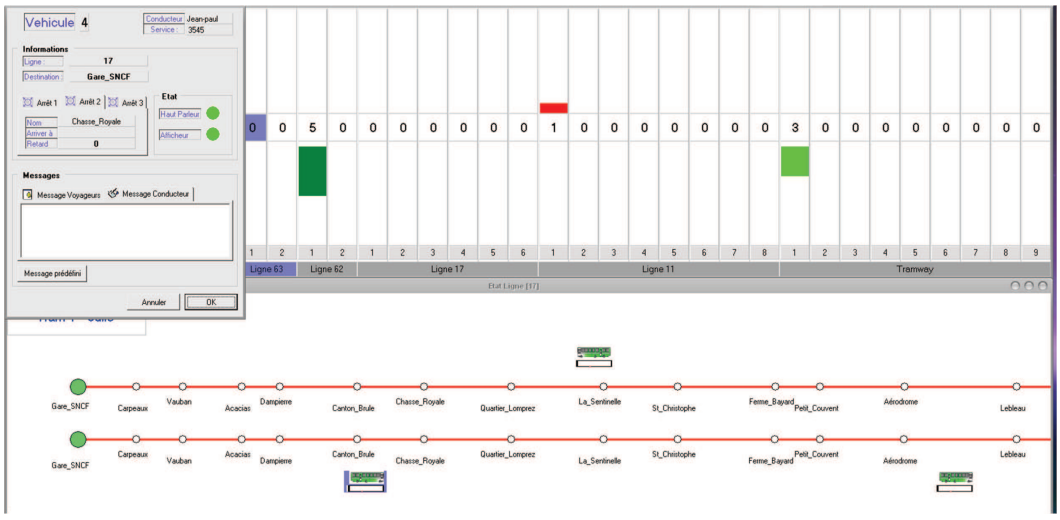



FIGURE 6.2: Interface du SAI : Interface « *Agent_Message* », Interface « *Agent_Etat_Ligne* » et Interface « *Agent_Etat_Trafic* »

En outre, le SAI permet d'informer les passagers (que ce soit en attente dans une station ou bien dans un véhicule) des différentes perturbations ayant lieu sur le réseau. Notons

que le SAI est un système ayant une architecture à base d'agents, Figure 6.3. Les lecteurs intéressés peuvent consulter (Bonte *et al.*, 2007; Ezzedine *et al.*, 2008; Trabelsi, 2006; Tran, 2009).

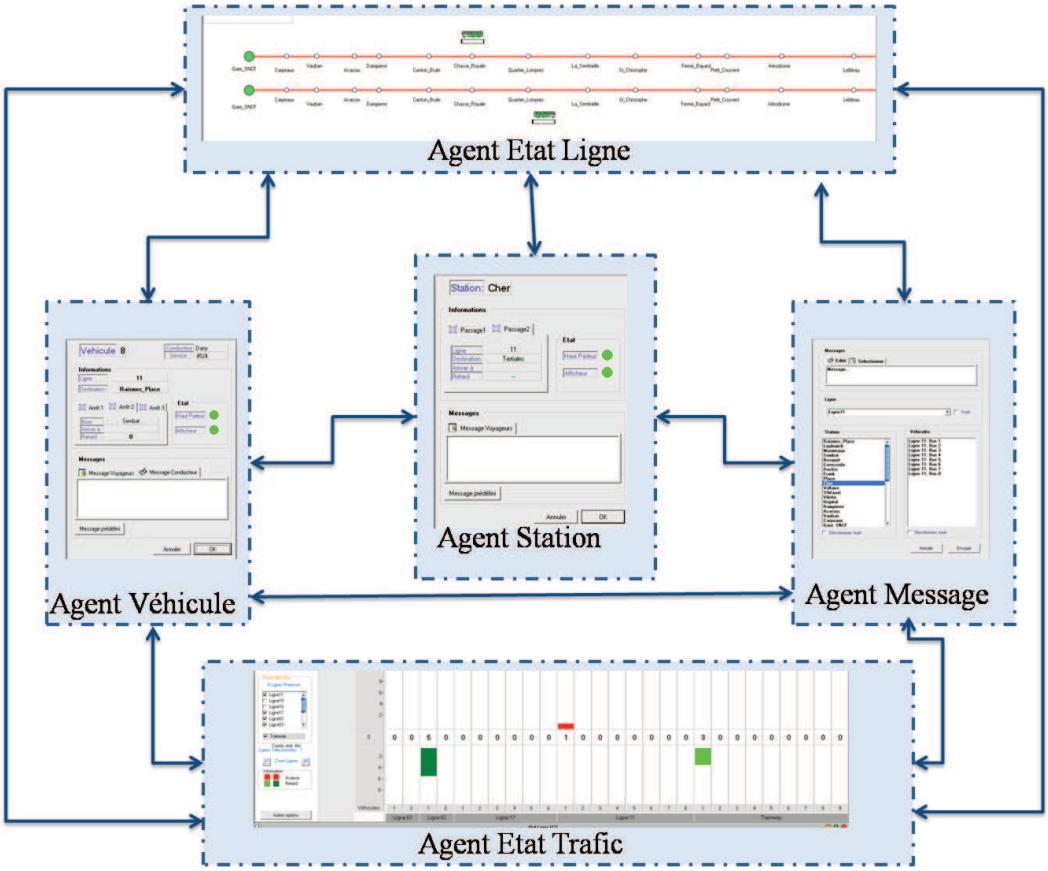



FIGURE 6.3: Architecture fonctionnelle du SAI

– Le SAD : ce système consiste à assister les régulateurs dans la prise de décision concernant la gestion des éventuelles perturbations sur le réseau de transport, Figure 6.4 (Bonte *et al.*, 2007).

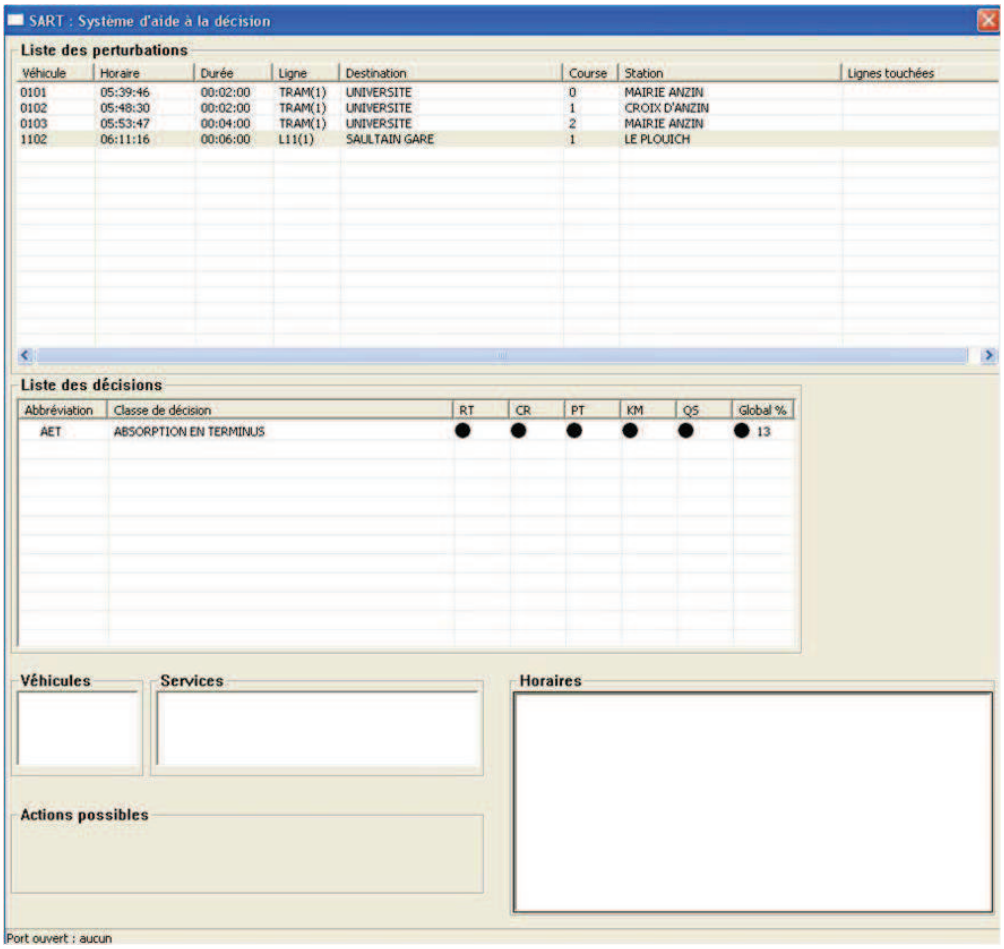



FIGURE 6.4: Interface du SAD

– Le SAE : ce système consiste à simuler un réseau de transport. Il est élaboré avec le logiciel « *Delmia Quest*[2] ». Cette simulation permet de visualiser en trois dimensions la circulation des différents véhicules dans un réseau de transport urbain.

2. Delmia Quest est un outil pour la modélisation, la simulation et l'analyse de processus industriel (Delmia, 2012).

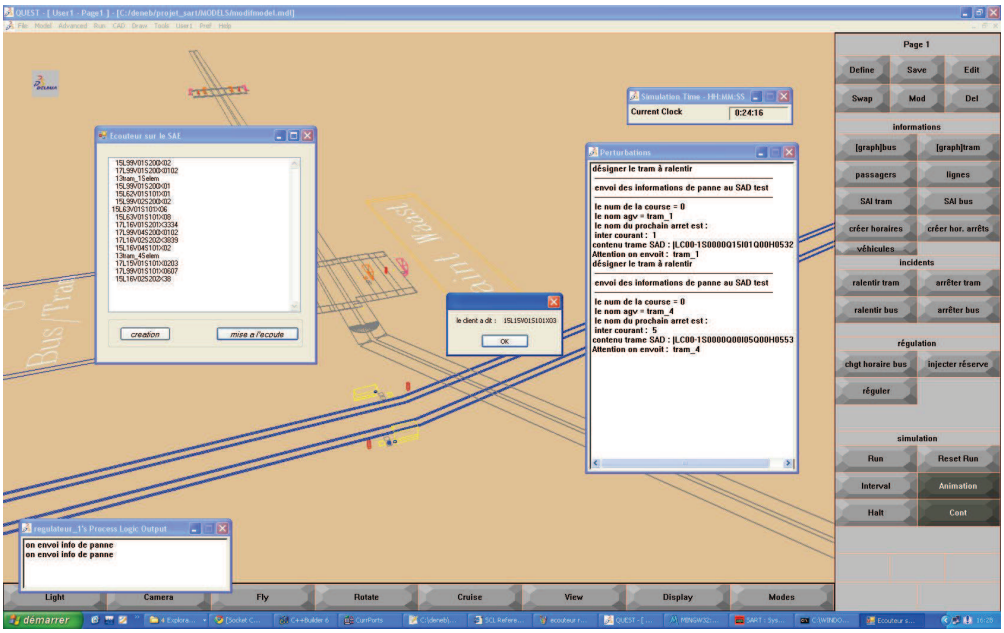



FIGURE 6.5: Interface du SAE

Dans cette section, nous allons procéder à l'évaluation du SAI.

### 6.1.2 Processus d'évaluation

*RITA* est un framework se basant sur trois différentes techniques pour l'évaluation du système interactif : le mouchard électronique, le questionnaire et l'inspection de la conformité de l'interface par rapport à des règles ergonomiques.

#### 6.1.2.1 Inspection de la cohérence des interfaces graphiques par rapport à des règles ergonomiques

Lors de l'inspection de la cohérence des règles ergonomiques, nous allons nous baser sur un ensemble de règles (cf. Annexe A). L'inspection des interfaces du système s'effectue interface par interface. Le système évalué dispose de 25 vues d'interfaces.

#### 6.1.2.2 Capture de l'interaction avec des utilisateurs

##### Choix des utilisateurs

Lors des évaluations des systèmes interactifs, l'évaluateur fait appel généralement à une dizaine de sujets par profil utilisateur (Virzi, 1990, 1992). Dans notre évaluation, nous allons nous restreindre à une seule catégorie de profils d'utilisateurs. Dans le but de neutraliser des effets parasites suite au choix des sujets, les sujets doivent répondre à

certaines conditions (Dumas et Redish, 1993) : acuité visuelle normale ou corrigée, même sexe, âgés de plus de 20 ans, etc. Jakob Nielsen affirme que les meilleurs résultats proviennent d'un test avec pas plus de cinq utilisateurs. En effet, il a montré avec Tom Landauer (Nielsen et Landauer, 1993) que le nombre de problèmes d'utilisabilité découverts avec un test à « $n$ » utilisateur(s) est donné suivant l'équation suivante : $N = (1-(1-L))^n$ ; (N est le nombre total de problèmes d'utilisabilité et L est le nombre de problèmes d'utilisabilité découverts avec un seul utilisateur). Nielsen et Landauer estiment que la valeur moyenne de L est de 31%. Pour cette valeur, nous obtenons la courbe (Figure 6.6) de l'évolution du nombre de problèmes découverts en fonction du nombre d'utilisateurs expérimentaux.

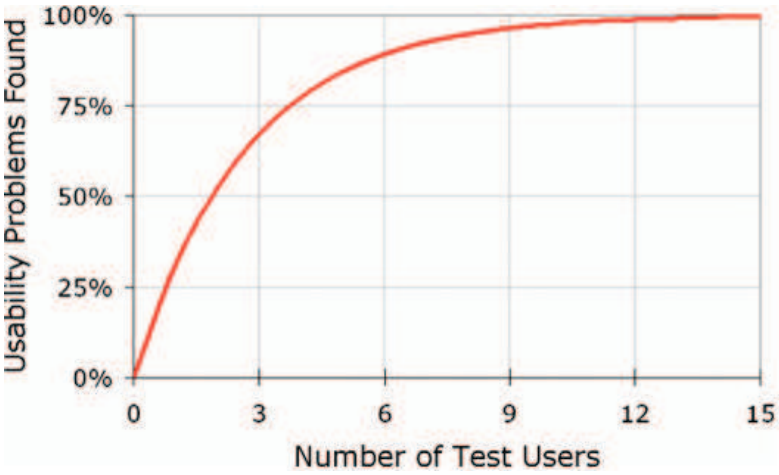



FIGURE 6.6: Evolution du nombre de problèmes d'utilisabilité découverts en fonction du nombre d'utilisateurs (Nielsen et Landauer, 1993)

Nielsen favorise le fait d'effectuer plusieurs petits tests plutôt qu'effectuer un seul test avec un grand nombre d'utilisateurs. Ceci permet une évaluation itérative de l'utilisabilité (Nielsen, 1993).

Les recommandations sur le choix du nombre de sujets sont très abondantes et variées (Hwang et Salvendy, 2010; Haynes *et al.*, 2004; Kwon *et al.*, 2007; Rubin et Chisnell, 2008; Whiting *et al.*, 2008; Dumas et Redish, 1993). Nous estimons que leur nombre est relatif à l'évaluation proposée (questionnaire, statistiques ordinaires, etc.) De plus, le nombre dépend étroitement de la nature du système à évaluer. A titre d'exemple, quand un système à évaluer est censé être utilisé par 1000 personnes, l'évaluateur ne doit pas procéder à l'évaluation avec un échantillon de quatre personnes. Par contre, une centaine d'utilisateurs peut être considérée comme étant représentative de la population. De plus, le nombre d'utilisateurs dépend aussi de la nature du système à évaluer (i.e. destiné au grand public ou bien système métier).

Dans notre évaluation, nous allons faire appel à une dizaine d'utilisateurs. Ces utilisateurs ont comme profils d'étudiants à l'*Université de Valenciennes et du Hainaut-Cambrésis*. Ils sont âgés entre 20 et 30 ans. Leur niveau d'expertise varie essentiellement entre étude en sciences informatiques et en sciences du sport.

**Scénario d'expérimentation**

Lors de la session d'interaction, l'utilisateur joue le rôle d'un régulateur de réseau. Il dispose d'une simulation d'un réseau qui connait des fortes perturbations au niveau de la circulation des véhicules dans le réseau. Chaque fois où il y a une perturbation dans le réseau (le passage d'un véhicule connaît du retard ou bien de l'avance par rapport à son horaire prévu de passage dans une station), le régulateur doit avertir le chauffeur et lui donner des consignes pour que le véhicule soit en accord avec les horaires prévus de passage dans les différentes stations du réseau. La communication avec le chauffeur s'effectue à l'aide de l'envoi de messages au conducteur du véhicule grâce à l'interface « *Agent_Message* ».

Notons qu'un régulateur peut connaitre essentiellement deux situations de travail lors de la supervision d'un réseau de transport :

– **Des situations normales**. Ces situations correspondent à un fonctionnement idéal du réseau de transport. Ce réseau ne connait pas de perturbations qui puissent engendrer des retards ou des avances des différentes véhicules du réseau par rapport aux horaires prévus de passage dans les stations. Le rôle du superviseur dans ce cas se restreint à suivre de prés le réseau.
– **Des situations perturbantes**. Il s'agit de situations pendant lesquelles le réseau connait des perturbations en matière d'avance et de retard de ses véhicules. Certains véhicules (Tramway ou Bus), pour diverses raisons, ne sont pas conformes aux horaires de passages prévus dans les stations. Le superviseur doit alors agir via le SAI en envoyant des messages aux conducteurs que ce soit pour accélérer ou bien ralentir la vitesse des véhicules et aux passagers pour les avertir d'un éventuel changement d'horaire. Le scénario prévu est marqué essentiellement par :
  - une fréquence élevée de perturbations sur le réseau ;
  - un chevauchement de plusieurs perturbations instantanées ;
  - toutes les lignes sont affectées par les perturbations et,
  - le changement fréquent des valeurs de retard/avance des différents véhicules.

Dans cette évaluation, nous allons procéder à des sessions de travail correspondant à une situation perturbante afin d'inciter l'utilisateur à exécuter le plus de tâche. Il consiste à injecter 48 perturbations sur le réseau de transport.

**Dispositif expérimental**

Le dispositif expérimental se base sur trois composants, Figure 6.7 :

– Le système interactif à évaluer : Le SAI ;
– Un mouchard électronique (EISEval) : qui se chargera de la capture des actions exécutées par l'utilisateur sur l'interface lors de la session de l'interaction et les stockera dans une base de données (Tran, 2009) et,
– Un émetteur de perturbations : est chargé d'envoyer des trames au SAI. Ces trames contiennent des perturbations dans le réseau de transport. Cet émetteur permet à l'utilisateur de créer des perturbations ou encore de charger un scénario préalablement défini et puis de transférer ces perturbations au SAI. Autrement dit, le rôle de cet émetteur est d'injecter des perturbations dans le réseau visualisé à l'aide du SAI.

La communication entre les trois composants s'effectue via le réseau TCP/IP par l'envoi de messages à travers les sockets.

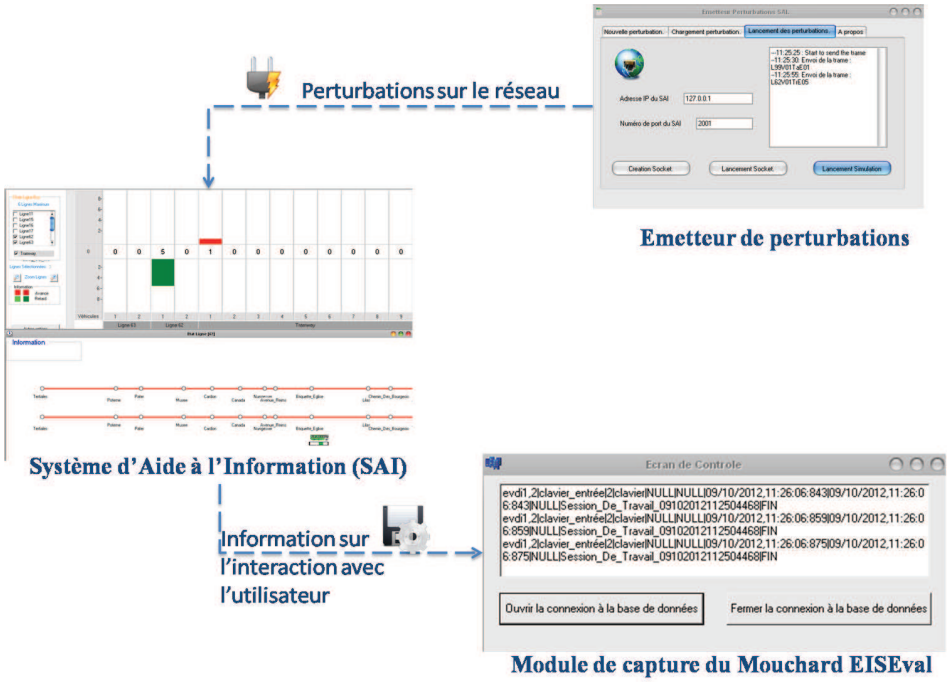



FIGURE 6.7: Dispositif expérimental

#### 6.1.2.3 Questionnaire

Généralement, avant de procéder à une évaluation expérimentale, les utilisateurs du système doivent répondre à un ensemble de questions afin de recueillir des informations sur eux tel que le nom, formation, niveau d'expertise, âge, etc. Ces questions visent à classer les utilisateurs suivant des profils. Dans notre évaluation, nous avons opté pour effectuer ces questions à la fin de la session d'interaction dans le but de les poser avec un autre ensemble de questions portant sur l'utilisation du SAI. Ces questions sont exposées dans l'Annexe B. Le recueil des réponses de l'utilisateur se fait via une interface HTML dans une base de données « *MS Access 2007* », Figure 6.8.

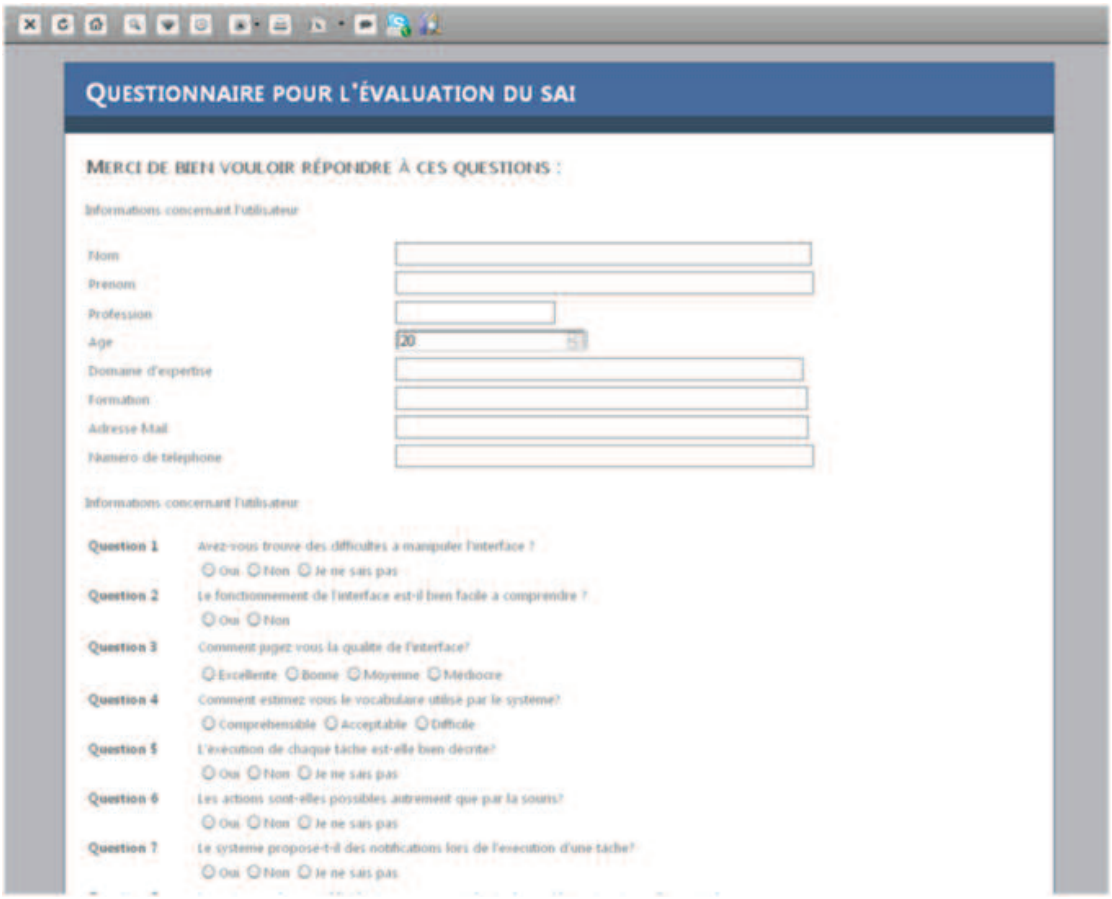



FIGURE 6.8: Une capture écran du questionnaire proposé pour les utilisateurs

### 6.1.3 Résultats

#### 6.1.3.1 Inspection de la cohérence de l'interface graphique du SAI par rapport à des règles ergonomiques

L'inspection de la conformité de l'ensemble d'interface du SAI a souligné un ensemble de problèmes d'utilisabilité liés à la conception de ces interfaces. Ces problèmes sont illustrés dans le tableau 6.1 ci-après.

| **Aspect inspecté** | **Description** | **Interfaces non conformes** |
|---|---|---|
| Nombre de couleurs utilisées pour les fontes d'écritures | Le concepteur s'est restreint à deux couleurs pour exprimer les différentes informations à l'utilisateur. | Etat_global, Etat_ligne 11, Etat_ligne 15, Etat_ligne 17, Etat_ligne62, Etat_ligne 63, Etat_Tramway, Etat_Trafic, Message, Alerte, ClienSocket, Etat_trafic, Cons, Fenêtre_Station, Fenêtre_Véhicule et Informations. |
| Contrôle explicite | Absence de bouton ou de composant permettant de remettre le système dans un état antérieur | Etat global, Etat_Ligne 11, Etat_Ligne 15, Etat_Ligne17, Etat_Ligne62, Etat_Ligne63 et Etat_trafic |
| Guidage | Les champs de texte ne présentent pas de valeur par défaut ou encore des indications | Etat_Ligne 11, Etat_Ligne 15, Etat_Ligne 62, Etat_Ligne63, Etat_Tramway. |
| Densité globale | Les interfaces sont surchargées de composants graphiques et par conséquent l'utilisateur trouve du mal à identifier les différentes informations. | Etat_global, Fenêtre_véhicule, Fenêtre_station, message et alerte |
| Absence d'indication | L'interface ne présente pas assez d'indications pour l'utilisation | Etat_global |
| Taille de la fonte d'écriture utilisée | La taille de la fonte utilisée pour afficher les informations à l'utilisateur doit être lisible et pas très grande. | Etat global, Etat_ligne 15, Etat_ligne11, Etat_ligne 17, Etat_ligne 62, Etat_ligne 63, Etat_Tramway, Etat_Trafic, Message, Alerte, Fenêtre_Station, Fenêtre_Véhicule, Informations, Lancement et Scénario |
| Couleur de l'arrière plan | L'arrière plan de l'interface ne doit comporter d'image ou encore de font coloré. | Etat_global, Etat ligne_15, Etat_ligne 17, Etat_ligne 62, Etat_ligne 63, Etat_Tramway, Etat_Trafic, Message, Alerte, Etat_trafic, Fenêtre_Station, Fenêtre_Véhicule, Informations et Scénario |

TABLE 6.1: Synthèse de résultat de l'inspection de la conformité de l'interface graphique du SAI par rapport à des règles ergonomiques

#### 6.1.3.2 Interaction

Le mouchard EISEval a permis de capturer des données relatives à l'interaction avec des utilisateurs. Grâce au module 3-1, nous déduisons que les dix utilisateurs ont passé en moyenne 16 minutes et 07 secondes et ont exécuté en moyenne 32 tâches, Tableau 6.2

| Utilisateur | Temps d'exécution | | Nombre de tâches exécutées | |
|---|---|---|---|---|
| | Temps réalisé | Ecart par rapport à la moyenne | Nombre d'actions exécutées | Ecart par rapport à la moyenne |
| 1 | 13 :34 | -11.87 | 31 | -09.68 |
| 2 | 18 :12 | +08.88 | 34 | +09.41 |
| 3 | 14 :12 | -11.34 | 29 | -11.03 |
| 4 | 15 :48 | -10.20 | 37 | +08.64 |
| 5 | 18 :04 | +08.92 | 26 | -12.30 |
| 6 | 16 :19 | +09.87 | 48 | +06.66 |
| 7 | 16 :50 | +09.57 | 33 | +09.69 |
| 8 | 14 :48 | -10.88 | 39 | +08.20 |
| 9 | 17 :02 | +09.46 | 32 | 00.00 |
| 10 | 16 :28 | +09.78 | 18 | -17.77 |
| Moyenne | 16 :07 | | 32 | |

TABLE 6.2: Synthèse sur les sessions d'interaction de dix utilisateurs avec le SAI

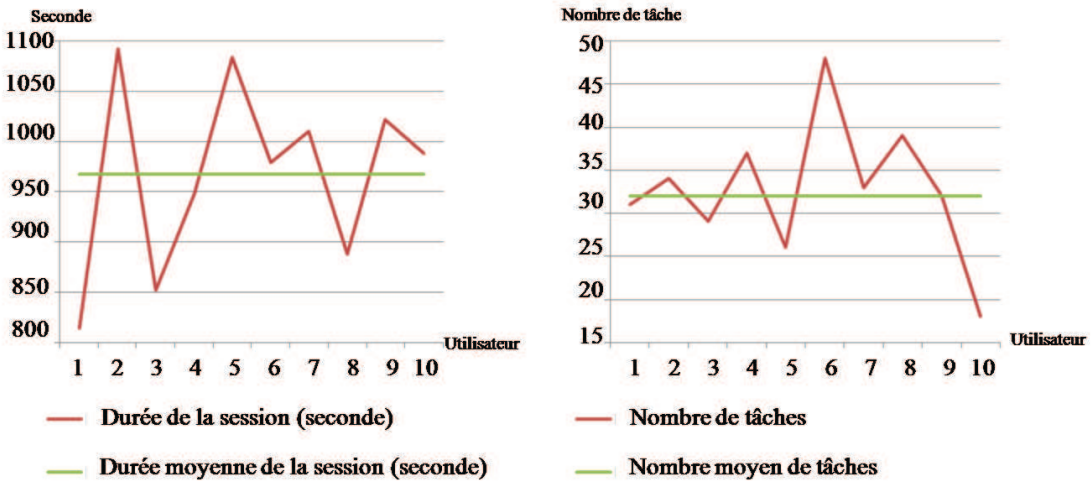



FIGURE 6.9: Courbe illustrant une synthèse sur les sessions d'interaction de dix utilisateurs avec le SAI.

#### 6.1.3.3 Questionnaire

A la fin de la session d'interaction, un questionnaire est proposé aux utilisateurs. Les résultats témoignent essentiellement que :

– les utilisateurs sont persuadés qu'en fin d'interaction ils ont pu réaliser les tâches demandées (neuf utilisateurs parmi dix) ;

– la majorité des utilisateurs s'attendait à avoir davantage d'informations au niveau de l'interface du SAI (sept utilisateurs parmi dix) ;
– les utilisateurs n'apprécient pas beaucoup la qualité de l'interface présentée (un seul la trouve excellente et deux utilisateurs l'ont estimé de bonne qualité et quant aux autres ils estiment qu'elle n'est pas de bonne qualité) ;
– le vocabulaire du système est accepté globalement par les utilisateurs (sept utilisateurs parmi dix) ;
– les utilisateurs proposent un meilleur guidage dans l'interface du fait que huit utilisateurs parmi dix estiment que les tâches ne sont pas bien décrites par le système ;
– la majorité des utilisateurs du système n'ont pas vérifié si une documentation ou encore de l'aide est proposée par le système (six utilisateurs parmi dix ont répondu qu'ils ne savent pas si elle existe ou non) et,
– les utilisateurs estiment que le système est bien facile à utiliser et à apprendre (huit utilisateurs parmi dix).

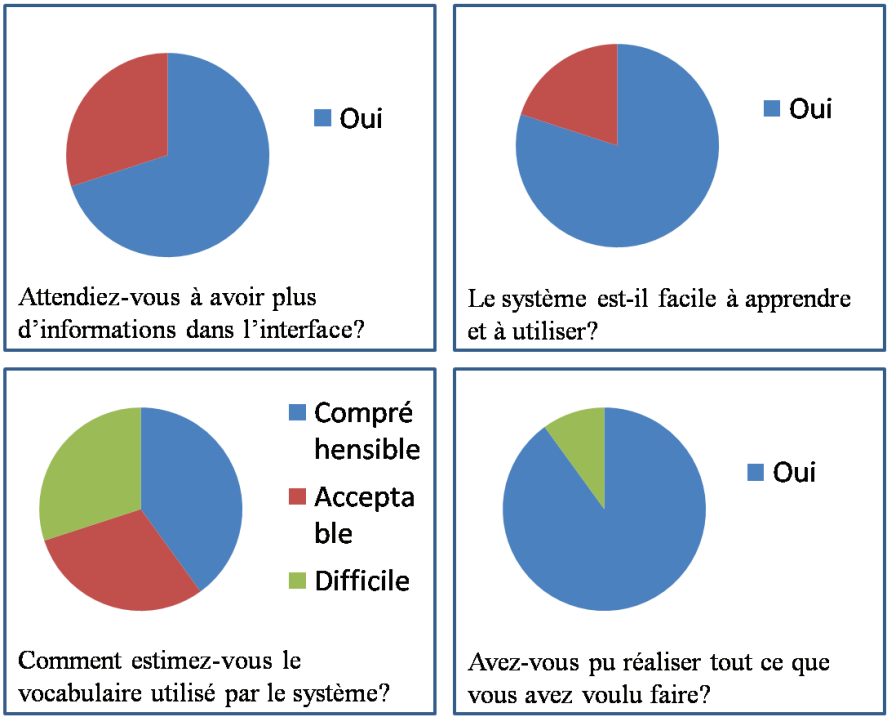



FIGURE 6.10: Exemple de résultats du questionnaire

### 6.1.4 Synthèse des résultats et discussion

Les résultats préliminaires des données capturées lors de l'interaction entre le SAI et les utilisateurs montrent que la majorité des utilisateurs n'ont pas accompli le même nombre de tâches avec le système. Notons que ces utilisateurs ont été soumis aux mêmes perturbations sur le réseau (48 perturbations). Ainsi, nous pourrons conclure qu'ils n'ont

pas réussi à gérer l'ensemble des perturbations sur le réseau. Ceci peut s'expliquer notamment par le fait que tous les utilisateurs n'ont pas fait attention au niveau interface « *Agent_Etat_Trafic* ». Dans cette interface, ils peuvent sélectionner et afficher les lignes qu'ils désirent afficher dans le baromètre. Cette interface utilisateur, par défaut, ne propose pas toutes les lignes au niveau de l'affichage. Par suite, l'utilisateur devrait activer toutes les lignes pour gérer l'ensemble du réseau. De plus, nous avons constaté que les utilisateurs font une confusion au niveau de l'interface « *Agent Message* » entre les messages destinés aux conducteurs et les messages destinés aux voyageurs du fait qu'il n'existe aucun mécanisme permettant de vérifier le destinataire du message avant l'envoi. D'ailleurs, durant l'expérimentation, nous avons constaté que les utilisateurs n'arrivent plus à un certain moment (là où il y a un chevauchement important de perturbations) à distinguer entre les véhicules auxquels ils ont envoyé des messages et les véhicules auxquels ils n'ont pas encore envoyé de messages. D'autre part, nous avons constaté que plusieurs utilisateurs confondaient entre le retard et l'avance des véhicules du réseau par rapport à leur horaire de passage. Cette confusion est due au choix des couleurs pour schématiser les perturbations (la couleur rouge pour l'avance et la couleur verte pour le retard). En effet, certains régulateurs ne traitent que le cas de l'avance du fait qu'ils croyaient ce qui est en vert est tout à fait normal (et le rouge anormal).

Puis, lors du changement de la valeur d'une perturbation, aucun message n'est envoyé à l'utilisateur. Par exemple, quand la valeur d'un retard passe de deux minutes à une seule minute, l'évaluateur n'a pas de mécanismes indiquant ce changement. Par suite, ce nouveau retard reste non traité par le régulateur.

Au niveau de l'inspection de la conformité de l'interface par rapport à des règles ergonomiques et du questionnaire dédié à des interfaces utilisateur, nous nous basons sur la norme l'ISO 9241-111 [3] (ISO, 1998)). Ces principes sont :

- **Clarté** : le contenu doit s'afficher clairement à l'utilisateur,
- **Discriminabilité** [4] : l'utilisateur doit être capable de distinguer facilement entre les différentes informations présentées sur l'interface.
- **Concision** : l'interface utilisateur ne doit pas être chargée avec des informations inutiles. Ainsi, elle ne doit comporter que les informations nécessaires.
- **Cohérence** : les informations doivent être cohérentes tout au long des interfaces utilisateur.
- **Détectabilité** : les informations sont codées de telle façon qu'elles soient facilement détectables et repérables par l'utilisateur. Autrement dit, les informations doivent être représentées avec la meilleure façon et au bon endroit.

3. Cette norme est publiée la première fois en 1998 sous la référence ISO 9241-12. Ensuite, en 2008 elle a été mise à jour et intitulée dans sa nouvelle version ISO 9241-111. Elle porte essentiellement sur la présentation des interfaces utilisateurs. Elle s'étend sur sept principes de base (ISO, 1998).

4. Traduction du terme en anglais « *discriminabilty* ».

| Principes | Questions et Règles ergonomiques associées |
|---|---|
| Clarté | Q3, Q17 et Q18 |
| Discriminabilité | R6, Q6 et Q7 |
| Concision | R5, R7 et R8 |
| Cohérence | Q9, Q12 et Q13 |
| Détectabilité | R6 et Q16 |
| Lisibilité | R2, R3, R4, R7 et R8 |
| Compréhension | R9, R10, R11, Q1, Q4, Q8, Q10 et Q11 |

TABLE 6.3: Association des règles ergonomiques et questions par rapport aux principes de la norme ISO 9241-111 (ISO, 1998)

– **Lisibilité** : les informations fournies doivent être visibles à l'utilisateur et faciles à lire.

L'évaluation a montré que les informations présentées aux utilisateurs sont clairement présentées. Généralement, l'utilisateur peut facilement distinguer entre les différentes informations. Néanmoins, lors de l'expérimentation avec l'utilisateur nous avons détecté que l'utilisateur peut retrouver des confusions au niveau de la distinction entre retard et avance (dû au choix des couleurs verte ou rouge) et au niveau du destinataire de message (conducteur ou passager). La concision des *IU* est respectée. Les informations présentées à l'utilisateur sont celles qui sont nécessaires pour le bon déroulement de la supervision du réseau de transport. La cohérence est aussi respectée sauf pour la couleur verte et la couleur rouge (comme mentionné au niveau du critère de l'discriminabilité). La détectabilité est aussi respectée. Par contre, au niveau du critère de lisibilité, les règles ergonomiques inspectées soulèvent certains problèmes. Tout d'abord, le concepteur s'est contenté de l'usage d'une seule couleur. De plus, certaines interfaces présentées sont assez denses et ainsi elles présentent des problèmes de lisibilité.

Afin de remédier aux différents problèmes d'utilisation détectés (ou encore éventuels problèmes), nous proposons essentiellement d'ajouter une interface permettant de consulter et de gérer l'historique des messages envoyés. De plus, nous proposons d'ajouter des notifications sur l'interface des perturbations traitées pour que l'utilisateur ne soit pas confus entre plusieurs perturbations.

De plus, l'interface évaluée ne présente aucun mécanisme visant à prévenir l'utilisateur de commettre des erreurs (se tromper de véhicule pour l'envoi de message par exemple). Les actions que l'utilisateur peut exécuter doivent être confirmées avant leur exécution. Outre cela, l'interface ne permet pas d'annuler les dernières actions afin de revenir à l'état antérieur. Par la suite, l'utilisateur ne peut pas rectifier ses actions erronées. Par ailleurs, le régulateur ne perçoit pas de confirmation de l'envoi de son message.

Ainsi, nous proposons une nouvelle maquette du SAI afin de remédier aux problèmes d'utilité et d'utilisabilité détectés dans l'ancienne maquette, Figure 6.11. Dans cette

interface, nous avons ajouté un tableau pour l'affichage de l'historique de l'ensemble des messages envoyés par le régulateur. De plus, nous proposons d'ajouter un icône auprès des retards traités par le régulateur. Outre cela, nous proposons aussi de fournir la possibilité au régulateur d'envoyer le message que ce soit au conducteur ou encore aux passagers par le clic sur la barre associé au véhicule correspondant dans le baromètre de l'agent « *Etat_Trafic* ». Par ailleurs, nous proposons d'ajouter sur l'agent état ligne des marqueurs de type alerte afin d'indiquer que la véhicule n'est pas conforme à son horaire de passage et qu'aucune action n'a été faite pour y remédier.

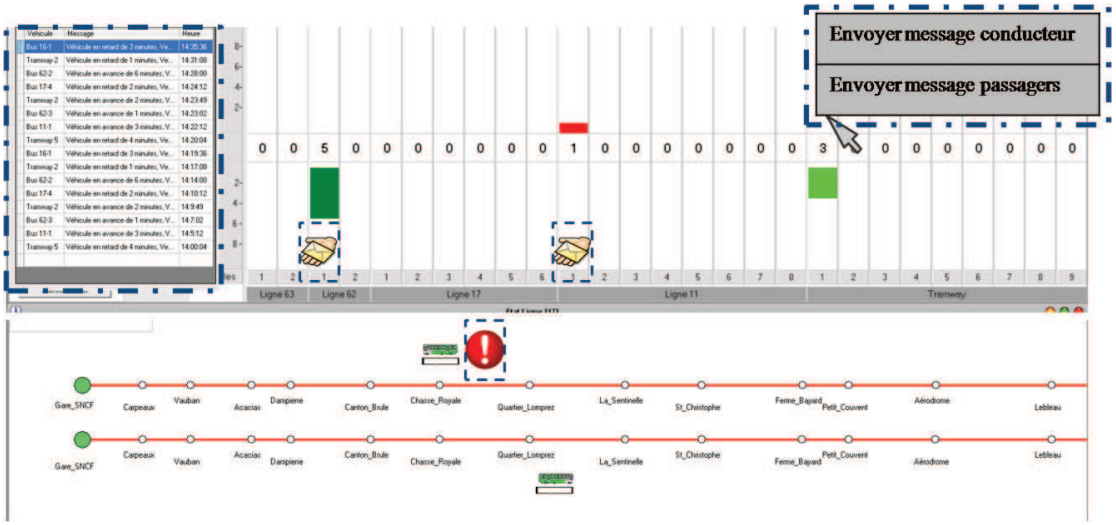



FIGURE 6.11: Proposition d'une maquette d'interface du SAI

## 6.2 Deuxième évaluation : évaluation des composants dédiés pour l'évaluation des interfaces utilisateur

### 6.2.1 Présentation du système à évaluer : SAI

Dans cette section, nous allons procéder à une évaluation expérimentale de la deuxième contribution présentée lors de ce mémoire. Cette évaluation porte sur un prototype du système évalué lors de la précédente section. Ce prototype a été conçu tout en prenant en considération l'ensemble des critiques émises précédemment. La Figure 6.12 illustre les améliorations apportées à l'IU du SAI. Ces améliorations sont encadrées en rouge pointillé dans cette figure :

– des indicateurs sur les icônes de véhicules mentionnant une perturbation ;
– un indicateur sur l'envoi de messages suite à une perturbation et,
– un agent interface chargé de stocker l'historique des messages envoyés par le superviseur ainsi que des différentes perturbations enregistrées sur le réseau de transport.

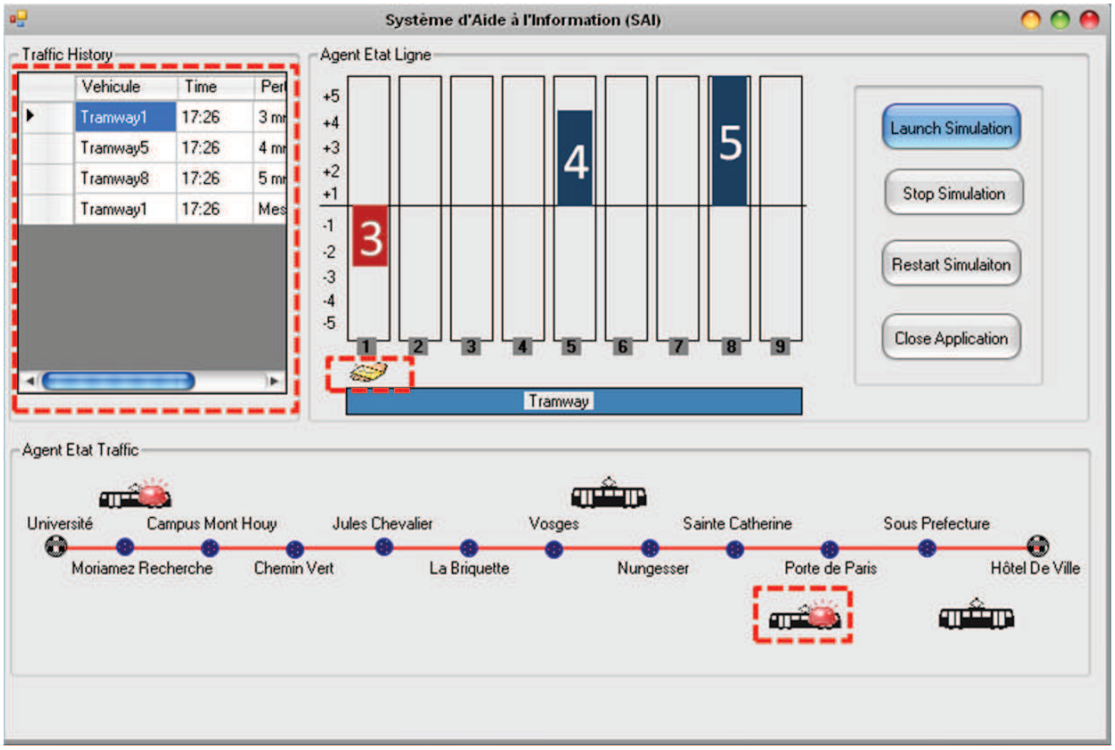



FIGURE 6.12: Une maquette d'un nouveau prototype du SAI.

Deux maquettes de ce prototype sont conçues afin d'effectuer cette évaluation. Elles sont conçues respectivement à l'aide des composants dédiés à l'inspection de l'utilisabilité et de l'utilité des IU.

### 6.2.2 Evaluation des composants dédiés pour l'inspection de l'utilisabilité des IU

#### 6.2.2.1 Conception/Evaluation

Cette évaluation est, comme mentionné dans le chapitre 5, est couplée à la phase de conception. L'évaluation ainsi consiste tout simplement à concevoir l'interface graphique à l'aide des composants proposés. Afin de procéder à cette conception/évaluation, nous modélisons un ensemble de règles ergonomiques sous forme de fichiers XML. Ces règles portent essentiellement sur : taille de l'écriture, couleur de l'écriture, fonte utilisée pour l'écriture, dimensions des images, dimensions des composants graphiques et nombre d'items dans un menu déroulant. Au fur et à mesure que le concepteur ajoute les composants à l'interface, les éventuelles erreurs de conception s'affichent. Une fois l'interface finalisée, le concepteur peut consulter un rapport global sur les règles non respectées et les améliorations envisagées. En effet, lors de la conception du SAI, le concepteur est avisé par un ensemble de recommandations proposées par les différents composants qui constituent l'interface, Figure 6.13. Les composants utilisés pour la composition graphique de l'interface sont : bouton, libellé, boîte à image, champ de texte et boîte combinée (Combo-Box).

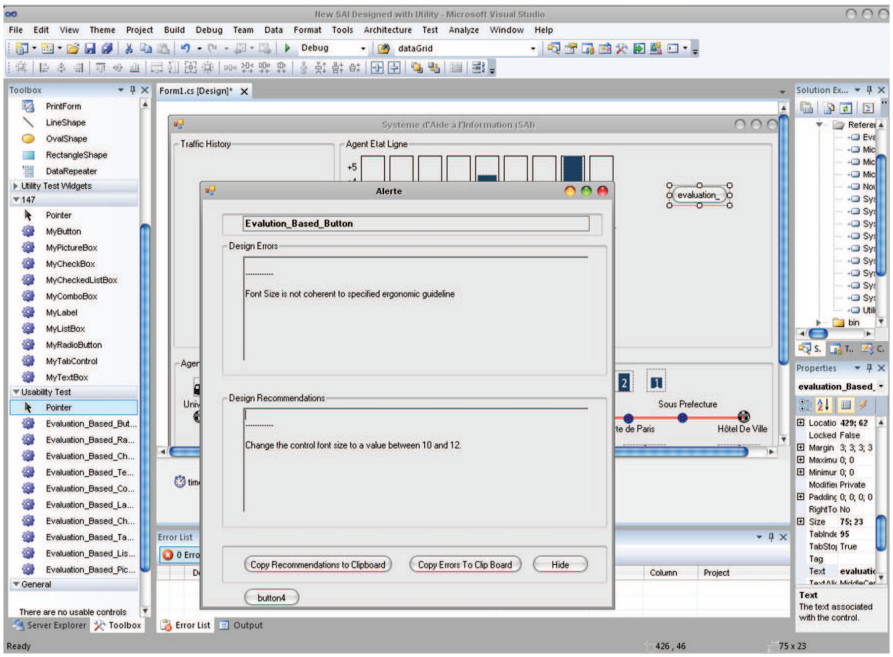



FIGURE 6.13: Une capture d'écran de la conception de l'interface utilisateur du SAI à l'aide des composants orientés inspection de l'utilité

| Composant | Fonte d'écriture | | | Couleur arrière-plan | Dimensions | | Valeur par défaut |
|---|---|---|---|---|---|---|---|
| | Taille | Couleur | Police | | Image | Composant | |
| Bouton | x | x | x | x | | x | |
| Bouton Radio | x | x | x | x | | x | |
| Case à cocher | x | x | x | x | | x | |
| Libellé | x | x | x | x | | x | |
| Liste de choix | x | x | x | x | | x | |
| Onglet | x | | x | | | | |
| Liste déroulante de case à cocher | x | x | x | x | | x | |
| Boîte Image | | | | | x | | |
| Champ de texte | x | x | x | x | | x | x |

TABLE 6.4: Exemples d'aspects considérés pour l'inspection de l'utilisabilité par les composants graphiques dédiés pour l'évaluation

#### 6.2.2.2 Résultat de l'évaluation de l'IU de la nouvelle maquette du SAI

L'évaluation de l'interface *IU* de la nouvelle maquette du SAI n'a pas soulevé des problèmes majeurs d'utilisabilité. En effet, les erreurs de conception détectées s'articulent essentiellement autour de la fonte d'écriture adoptée par l'IDE « *MS Visual 2010* » est la « *Microsoft Sans Serif* » alors que la RE associée recommande l'utilisation de la fonte utilisée par le système d'exploitation. De plus, une des REs adoptées lors de l'évaluation, exige une taille d'écriture plus élevée (10 points) que celle utilisée par l'IDE (8.25 point).

### 6.2.3 Evaluation des composants dédiés pour l'inspection de l'utilité des *IU*

Dans cette sous-section, nous allons procéder à une évaluation de l'*IU* de la maquette du SAI, Figure 6.12, à l'aide de la deuxième famille des composants orientés évaluation des *IU*. L'évaluation consiste à concevoir l'interface graphique avec les composants proposés dans Chapitre 5 (Section 5.3). Ensuite, les tâches possibles avec cette interface sont modélisées suivant la notation CTT. Par la suite, le modèle de référence est établi. Ensuite, nous procéderons à des sessions d'interactions avec des utilisateurs. Ces sessions d'interaction nous amèneront à spécifier les différents modèles d'objet. Vers la fin, la

comparaison entre les deux types de modèle nous renseigne sur les tâches dont l'exécution soulève des problèmes d'utilisation.

#### 6.2.3.1 Conception de l'interface

#### 6.2.3.2 Modèles de tâches

Afin de tester expérimentalement les composants dédiés inspection de l'utilité avec une maquette du SAI, nous avons opté pour le choix de trois tâches seulement pour cette évaluation, Figure 6.14. Ces tâches sont :

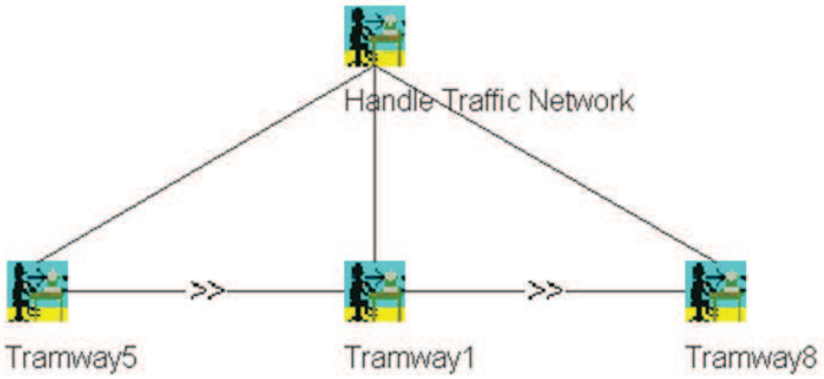



FIGURE 6.14: Modélisation des tâches requises pour l'utilisateur en CTT

- Envoi de message au conducteur du véhicule (Tramway 5) suite à une avance de quatre minutes ;
- Envoi de message au conducteur du véhicule (Tramway 1) suite à un retard de trois minutes et,
- Envoi de message au conducteur du véhicule (Tramway 8) suite à une avance de cinq minutes.

Chaque tâche est décomposée suivant quatre sous-tâches, Figure 6.16. La première sous-tâche consiste à cliquer sur l'icône associée au retard. Cette icône n'est qu'un composant graphique du type Boîte à Image (Picture Box). Ce clic permet d'afficher une fenêtre permettant d'envoyer les messages. Cette fenêtre permet de renseigner des informations (le nom du conducteur, la station desservie par le véhicule, la prochaine station, la ligne et la direction du véhicule), Figure 6.15. Ensuite, la seconde sous-tâche correspond à la spécification du destinataire par l'utilisateur (le superviseur). Pour s'y faire, l'utilisateur

choisit le destinataire à partir d'une boîte combinée (Combo-box). Les destinataires proposés du message sont : le conducteur, les passagers en attente dans une station et les passagers dans le véhicule. Puis, la troisième sous-tâche consiste à saisir le message à envoyer dans un champ de texte. Enfin, la quatrième sous-tâche correspond à la validation de l'envoi du message via l'appui sur un bouton (« *Send* »).

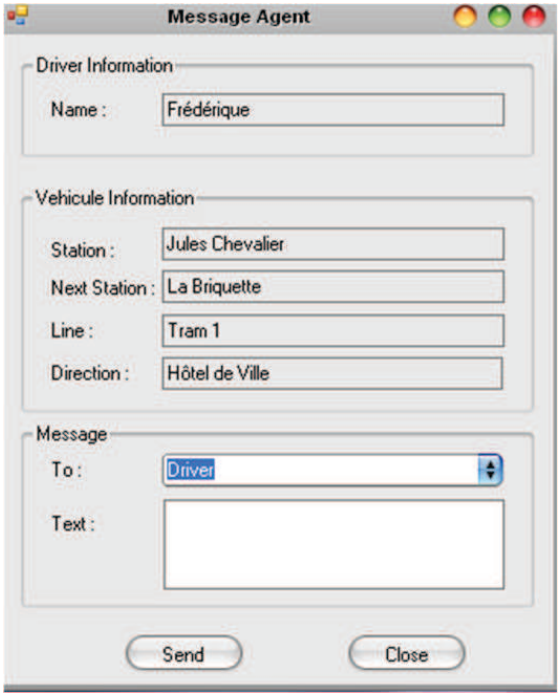



FIGURE 6.15: Capture d'écran de l'interface d'envoi de message avec le SAI

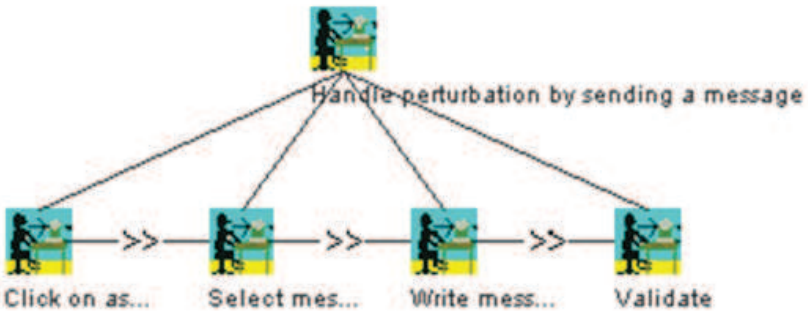



FIGURE 6.16: Modélisation de la décomposition de la tâche utilisateur en sous-tâches en CTT

#### 6.2.3.3 Scénario d'expérimentation

Le scénario expérimental consiste à demander à des utilisateurs de ce système, de gérer les perturbations ayant lieu sur le réseau via la maquette proposée.

### 6.2.4 Processus d'évaluation

#### 6.2.4.1 Dispositif expérimental

Comme mentionné dans le Chapitre 5, la solution proposée s'articule autour d'une architecture client-serveur. Ainsi, le dispositif expérimental est comme mentionné dans Figure 6.17. Dans ce dispositif l'évaluateur interagit avec le mouchard électronique dédié pour l'évaluation des interfaces utilisateurs et l'utilisateur avec l'interface graphique conçue à partir des composants orientés évaluation. Ces composants communiquent avec le mouchard via l'envoi de messages à l'aide de sockets pour communiquer les différentes informations concernant leur utilisation. Notons que les composants utilisés pour la composition de l'interface évaluée sont : boîte d'image, bouton, boîte combinée, champ de texte et libellé.

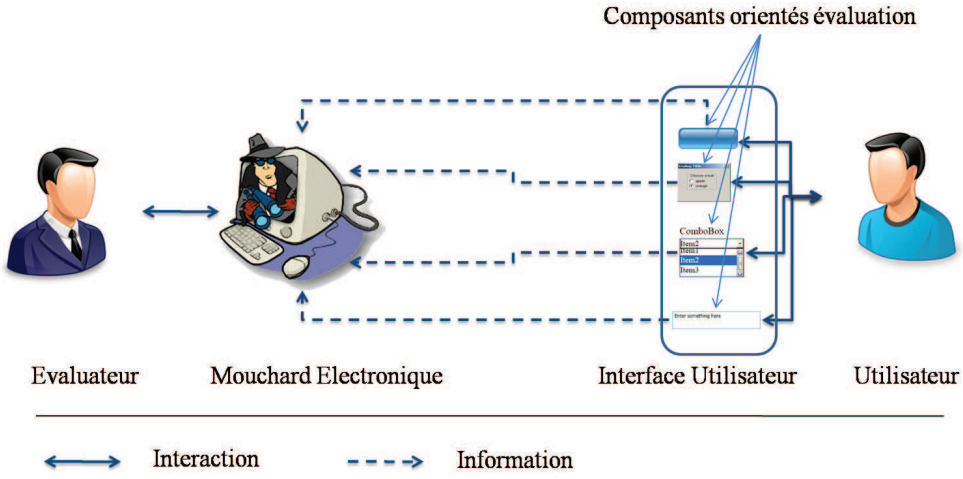



FIGURE 6.17: Dispositif expérimental

#### 6.2.4.2 Résultats de l'évaluation

L'évaluation a montré tout d'abord la facilité du processus d'évaluation avec les composants orientés inspection de l'utilité des interfaces utilisateur. Rappelons que l'objectif de cette section est non pas d'évaluer l'interface graphique mais plutôt de tester les fonctionnalités de ces composants. Le test de l'*IU* n'a pas soulevé de problèmes d'utilisation dans l'interface utilisateur. Ceci peut se traduire par la simplicité et la non complexité

de tâches exécutées. En effet, les utilisateurs ont pu exécuter les trois tâches requises (Gestion des perturbations du Tramway5, Tramway 1 et Tramway 8). Enfin d'évaluation, l'évaluateur dispose du taux de réussite de l'exécution des tâches par les différents utilisateurs ainsi que des séquences d'utilisation, Figure 6.18. Notons que l'une des limites

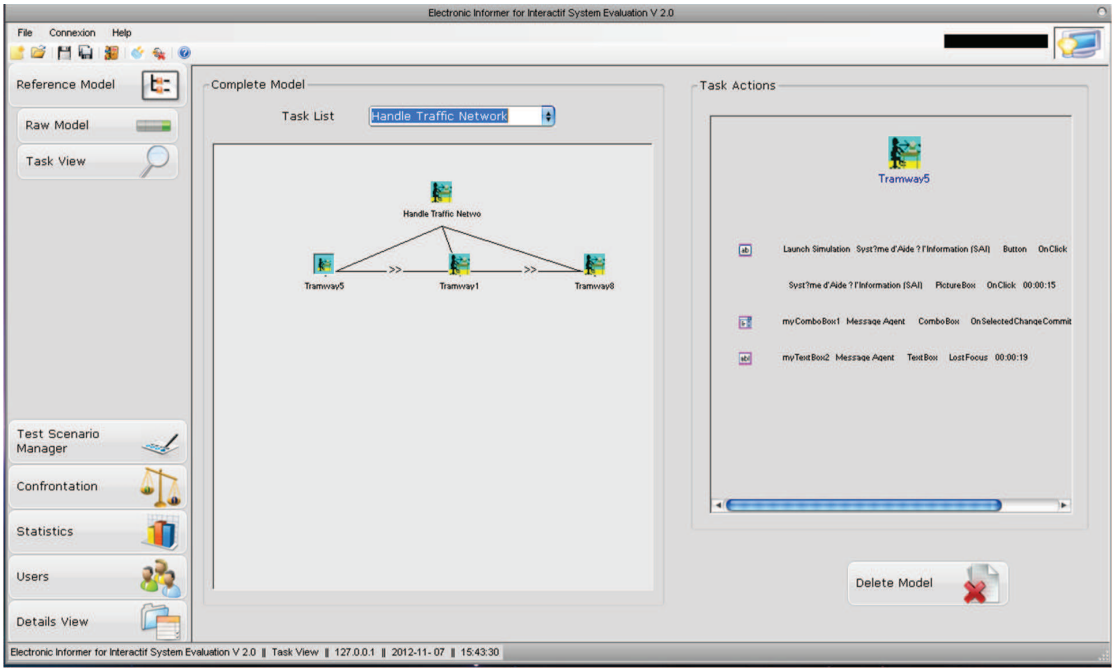



FIGURE 6.18: Aperçu d'un exemple de résultat de confrontation entre modèle de référence et modèle d'objet.

du mouchard proposé pour l'évaluation des *IU* est l'impossibilité de sauvegarder les actions exécutées et les résultats d'évaluation sous forme exploitable pour plus d'études et d'analyses. En effet, l'évaluateur se trouve ainsi restreint aux données numériques proposées par le mouchard. De plus le mouchard ne permet pas de spécifier précisément l'endroit des problèmes d'utilisation. En outre, les statistiques élaborées par le mouchard restent peu élaborées et nécessitent d'avantage de travail.

## 6.3 Discussion et synthèse

L'évaluation de la partie interactive des systèmes interactifs est une phase primordiale pour la validation des systèmes interactifs. Comme cité lors du chapitre 1, les travaux relatifs à l'évaluation des *IU* sont nombreux et variés. Les chapitres 2 et 3 ont mis l'accent sur la diversité d'outils d'évaluation des *IU*. Nous avons vu que la majorité d'entre eux procédent à l'évaluation que lors des dernières phases de conception des SI. De plus, ces outils se basent généralement sur une seule technique pour le recueil des données pour l'évaluation.

Dans ce mémoire nous avons présenté deux approches pour l'évaluation des systèmes interactifs. La première consiste dans un framework pour l'évaluation des *IU*. Ce framework a été baptisé *RITA*. Son principal avantage est le fait qu'il est :

– générique pour l'évaluation des systèmes interactifs ;
– automatique lors de la capture et l'analyse des informations pour l'évaluation des *IU* ;
– basé sur une architecture modulaire qui permet sa configuration lors des processus pour l'évaluation des différentes *IU* ;
– multi-techniques. Il se base sur trois différentes techniques pour l'évaluation et par suite il couvre un spectre plus large d'aspects pris en considération lors de l'évaluation : présentation statique de l'interface, interaction entre l'utilisateur et le SI et aussi l'appréciation de l'utilisateur vis-à-vis de l'interface évaluée ;
– Flexible au niveau des règles ergonomiques prises en considération lors de l'inspection de la qualité ergonomique des interfaces utilisateurs et,
– Multicritères. Il permet d'inspecter à la fois l'utilité et l'utilisabilité des interfaces utilisateurs.

Ce framework a été testé afin de le valider dans le cas d'un système de régulation de réseau de transport dans le cadre du projet CISIT-ISART. Lors de ce test, nous avons testé l'efficacité de ce framework. Ceci nous a permis de dégager un ensemble de limites. Ces limites sont essentiellement :

– La qualité et le format du rapport d'évaluation proposé reste néanmoins peu élaboré pour une meilleur exploitation par l'évaluateur ;
– la phase de critique de l'interface utilisateur reste au soin de l'évaluateur. En effet, le framework ne permet pas d'évaluer les interfaces utilisateur mais plutôt d'assister l'évaluateur lors de ce processus via la capture et l'analyse automatique des données pour l'évaluation ;
– la définition des règles ergonomique et leur exploitation reste applicable que pour les règles simples et classiques. De plus, il n'est pas possible d'aborder des règles qui sont liés à deux attributs différents du composant graphiques ;
– l'impossibilité d'évaluer les interfaces utilisateur de type Post-WIMP ;
– la modélisation de la tâche utilisateur reste néanmoins mal adaptée et n'applique aucun formalisme dédié à de telle modélisation des tâches utilisateurs. Cette modélisation permettrait ainsi de faciliter le processus d'analyse des tâches des utilisateurs.
– la phase de configuration du framework reste une tâche délicate et assez complexe lors de la pré-évaluation.

La seconde contribution concerne les composants graphiques pour la composition d'interfaces utilisateur. Ces composants sont orientés évaluation précoce des interfaces utilisateurs. Le point fort derrière ce concept réside dans le fait que ces composants impliquent des mécanismes pour l'évaluation des interfaces utilisateur. Ainsi, l'évaluation est effectuée au même temps que la conception de l'interface.

Ces composants s'articulent autour de deux familles. La première est dédiée pour l'inspection de l'utilisabilité dans les interfaces utilisateur. Les composants graphiques issus de cette famille comportent des mécanismes internes permettant de parcourir des fichiers XML. Ces fichiers modélisent des RE. De plus, ces composants incluent des mécanismes permettant de s'auto-inspecter la conformité par rapport à ces règles. Autrement dit, l'évaluation proposée est supportée depuis les premières phases de conception des SI ; c'est le cas de l'évaluation précoce. Rappelons qu'aucune spécificité d'évaluation n'est apparente à l'utilisateur et que leur utilisation est bien similaire à celle des composants fournis par les *EDI* de type *WYSIWYG*. Les outils permettant l'inspection de la qualité ergonomique des *IU* sont nombreux. La majorité d'entre eux propose une évaluation automatique consistant à identifier les aspects pouvant engendrer des problèmes d'utilisation de l'interface. Néanmoins, la correction de ces aspects peut requérir une seconde conception de l'interface pour remédier aux problèmes d'utilisabilité dans l'interface et parfois même de rectifier le noyau fonctionnel du SI pour réaliser ces améliorations. De telle opération est souvent coûteuse en termes de coût matériel, d'opérateurs humains et de temps. En revanche, les composants proposés dans ce mémoire évitent de tel « *gaspillage* » pour l'amélioration des *IU*.

La seconde famille concerne des composants qui peuvent être exploités pour la capture des interactions entre l'utilisateur et l'*IU* : les composants dédiés pour l'inspection de l'utilité des interfaces utilisateurs. Ces composants sont dotés de mécanismes permettant de communiquer à un mouchard électronique les informations concernant leur utilisation en temps réel via les sockets. Ce mouchard se base sur le concept énoncé par Senach (Senach, 1990). L'avantage de ces composants consiste dans le fait de supporter l'évaluation depuis les premières phases de conception de l'interface graphique. Ainsi la capture des données concernant l'évaluation d'une façon automatisée n'implique pas le recours à l'insertion de ligne de code dans le code source du système à évaluer comme dans la plus part des outils d'évaluation se basant sur le mouchard électronique. De plus, ces composants permettent l'évaluation des maquettes des interfaces graphiques. Autrement dit, l'évaluation se fait comme étant la validation d'une maquette avant d'entamer la phase de conception du système interactif.

Néanmoins ces composants n'existent que dans une première version qui reste à raffiner et à optimiser. En effet, la définition proposée des règles ergonomiques ne permet pas (comme dans la première contribution) de modéliser la majorité des règles ergonomiques. Elles ne concernent que les règles dites classiques qui n'inspectent que certains aspects tels que la police et les dimensions des composants graphiques ; ils ne concernent pas à titre d'exemple l'alignement et le regroupement des différents composants dans

l'interface. De plus, le rapport proposé en fin d'évaluation n'est qu'une collection d'erreurs détectées lors de la phase de conception/évaluation et les recommandations associées. Quant aux composants dédiés pour l'inspection de l'utilité des *IU*, ils adoptent le même concept pour l'évaluation à l'aide de mouchard (capture des interactions et leur analyse). L'évaluation consiste essentiellement en une comparaison entre différentes séquences d'interactions. Il est à noter que nous avons essayé de combiner entre les différents mécanismes (auto-inspection de la conformité par rapport à des règles ergonomiques et l'envoi d'information au mouchard) suivant une seule famille de composants. En revanche, les composants résultants témoignent d'une grande lenteur au niveau de l'utilisateur.

# Conclusion générale et Perspectives de recherche

## Conclusion Générale

L'IHM comme discipline du génie logiciel vise à concevoir, implémenter et évaluer des systèmes interactifs afin qu'ils puissent accomplir la meilleure communication possible avec l'utilisateur (Hewett, 1992). Beaucoup de travaux dans ce contexte ont été proposés pendant ces vingt dernières années (Vanderdonckt, 1994). Dans le présent mémoire, nous nous sommes intéressés à ceux relatifs à l'évaluation de la partie interactive des systèmes interactifs. Cet intérêt s'est accentué par le fait que cette partie est, elle seule, la partie apparente à l'utilisateur. En effet, les milliers de lignes de codes sont dissuadées par l'interface utilisateur (Meyer, 1997). Lors du premier chapitre de ce mémoire, nous avons présenté brièvement les principales techniques d'évaluation. Ensuite, dans le second chapitre, nous nous sommes focalisés sur les outils se basant sur des connaissances ergonomiques pour assurer l'évaluation des *IU*. Ensuite, lors du troisième chapitre, nous nous sommes intéressés à l'étude des outils se basant sur la capture et l'analyse des interactions entre l'utilisateur et l'interface utilisateur. Ces deux études ont souligné le nombre important des outils existant pour l'évaluation des *IU*. De plus, ces deux études ont soulevé un ensemble de limites relatives à ces outils d'évaluation. Tout d'abord, ces outils s'appliquent en grande majorité lors des phases finales du cycle de conception de systèmes interactifs. Ceci implique un coût élevé du processus d'évaluation proposé pour la correction des éventuelles incohérences au niveau de l'interface. Prenons à titre d'exemple le cas des outils basés sur l'inspection de la qualité ergonomiques comme technique d'évaluation. L'évaluation propose généralement un ensemble d'aspects de l'interface non conformes aux connaissances ergonomiques de

l'interface et même des modifications au niveau du noyau fonctionnel du système interactif. Ainsi, nous relevons l'importance de l'évaluation précoce au niveau du gain en coût en termes d'effort, de ressources matérielles et de temps. Ensuite, nous avons constaté que chaque outil se base généralement sur une seule technique d'évaluation. Or chaque technique s'adresse à un ensemble d'aspects distincts aux aspects pris en considération pour d'autres aspects. En effet, de nombreux auteurs recommandent de procéder à une évaluation basée sur différentes techniques afin de couvrir un spectre plus large d'aspects pris en considération lors de l'évaluation et par conséquent aboutir à de meilleurs résultats (Nielsen, 1993). Puis les outils existants proposent une évaluation automatisée lors de la capture et de l'analyse des données alors que la phase de critique est souvent délaissée aux soins de l'évaluateur. Ainsi, les résultats d'évaluation peuvent diverger d'un évaluateur à un autre et d'où les résultats sont moins fiables. L'évaluation relèvera ainsi, à des jugements objectifs de la part de l'évaluation. Enfin, les outils se basent sur des connaissances ergonomiques codant généralement ces connaissances dans leur propre moteur d'évaluation. Ainsi l'évaluateur se trouve restreint à l'ensemble de règles proposées par l'outil d'évaluation utilisé. Ces limites nous ont servi en termes de motivations pour l'évaluation de deux contributions qui sont présentées dans ce mémoire. La première s'articule autour d'un Framework générique, configurable et multi-méthodes pour l'évaluation de la partie interactive des systèmes interactifs. Ce Framework se base sur l'inspection de la qualité ergonomique, le mouchard électronique et le questionnaire comme étant des techniques d'évaluation des interfaces utilisateur. Ainsi, l'évaluation porte sur la présentation statique de l'interface, l'interaction proposée avec l'utilisateur et son appréciation vis-à-vis du système évalué. L'architecture modulaire de ce Framework permet sa configuration pour l'évaluation des différentes interfaces utilisateur. La deuxième contribution consiste en un ensemble de composants pour la composition graphique d'interface utilisateur en utilisant les environnements de développement intégré de type WYSIWYG. Autre que la fonctionnalité de composer les interfaces utilisateur, ces composants ont pour objectif de supporter l'évaluation depuis les premières phases de conception de systèmes interactifs. Ces composants sont catégorisés suivant deux familles ; la première concerne des composants ayant la spécificité de s'auto-inspecter la conformité par rapport à un ensemble de règles ergonomiques préalablement spécifié par l'évaluateur. Ils communiquent les incohérences ergonomiques au concepteur lors de la conception de l'interface utilisateur dans le but d'assister le concepteur à implémenter des interfaces cohérentes par rapport aux règles sélectionnées. Cette famille est dédiée, ainsi, à l'inspection de l'utilisabilité des interfaces utilisateur. La seconde famille vise à inspecter l'utilité des interfaces utilisateur. Pour s'y faire, elle opère avec un mouchard électronique dédié à cet usage. L'évaluation proposée s'effectue via la comparaison entre le séquencement d'actions prévu par l'évaluateur et les actions exécutées par les utilisateurs pour exécuter une tâche donnée. Cette comparaison est élaborée dans le but

d'identifier les tâches et les sous-tâches relevant des difficultés d'exécution et par conséquent d'en déduire les aspects de l'interface engendrant de telles difficultés d'utilisation. Ces deux contributions sont proposées dans une première version. L'évaluation et le test de ces contributions ont relevé un certain nombre de limites relatives à leur utilisation et leurs fonctionnalités. Dans un premier lieu, la qualité des rapports de l'évaluation requiert davantage d'amélioration. En effet, il serait nettement meilleur d'opter à des standards dédiés aux rapports d'évaluation.

De plus, une limite reste encore soulevée portant sur la qualité de l'interface. En effet, tout comme les outils existants, ces deux contributions ne permettent pas d'annoncer clairement sur la qualité des interfaces utilisateur (excellente, très bonne, bonne, moyenne, médiocre, etc.). Elles se restreignent à détecter et à présenter les différentes incohérences et les problèmes d'utilisation rencontrés lors de l'interaction.

## Perspectives de recherche

Dans ce mémoire, nous nous sommes intéressés à l'évaluation des *IHM*. Nous en avons proposées deux contributions. Nous proposons dans cette section des perspectives de recherche relatives à ces deux contributions et à la thématique de l'évaluation des *IHM*.

### Le Framework RITA

Nous avons vu que *RITA* propose des rapports de différentes formes (PDF, texte et Web). Néanmoins, le rapport n'obéit pas aux normes de rapports d'évaluation. En effet, ils existent quelques standards pour les rapports d'évaluation. Ces standards sont essentiellement EARL et RDL. EARL est un standard adopté pour la W3C. Il vise essentiellement à présenter le rapport comme étant quatre parties : Evaluateur, le sujet de test, le critère adopté lors du test et les résultats du test. Le second standard est proposé par Microsoft : RDL. Il consiste à proposer un schéma commun des rapports pour faciliter leur compréhension et exploitation et d'améliorer leur interopérabilité. Il est basé sur le langage de balisage XML. Ces deux normes sont dédiées à l'évaluation en général et ne prennent pas la spécificité de l'évaluation des *IHM*. En effet, il n'existe pas de normes ou bien de standards dédiés pour l'évaluation des *IHM*. Ainsi, nous proposons comme première perspective d'élaborer une modélisation de rapport d'évaluation des *IHM* et de l'exploiter pour le quatrième module concernant la génération de rapport d'évaluation.

De plus, nous nous proposons de quantifier la qualité des *IU* évaluées suivant des critères bien définis pour pouvoir, à titre d'exemple, élaborer des comparaisons entre les alternatives de conception. Le résultat d'évaluation serait ainsi plus pertinent et intéressent. En

effet, nous avons vu lors des chapitres 2 et 3 que les outils existants ne proposent généralement qu'une liste des problèmes d'utilité et d'utilisabilité. Ces outils ne permettent pas de donner un jugement claire sur l'interface (qualité : excellente, très bonne, moyenne, etc.). Ainsi, nous nous proposons de se baser sur des travaux liés à l'aide à la décision pour quantifier la qualité de l'interface et par conséquent pour proposer des jugements portant sur cette qualité.

Comme troisième perspective, nous nous proposons de déployer le Framework sous forme de service web afin de garantir une meilleur interopérabilité et une utilisation multiplateforme et de l'exploiter pour des différentes évaluations. Ainsi ce Framework sera muni d'une architecture orientée services.

Il est claire que les interfaces de type post-WIMP sont de plus en plus répondus au niveau de l'usage quotidien et leur utilisation est de plus en plus fréquente (interface tactile, interface tangible, etc.). Néanmoins, il n'existe pas encore de modèle, de technique ou d'outil pour valider et tester l'interface qu'ils proposent. Nous proposons d'augmenter le spectre de fonctionnalités de *RITA* en intégrant l'évaluation des interfaces de type post-WIMP. Ces interfaces utilisateurs ont leur spécificité et sont différent avec les autres interfaces.

### Composants orientés évaluation

Nous avons présenté deux familles de composants graphiques. Nous estimons qu'il serait plus intéressant d'englober le tout sous forme d'une seule famille de composants. Ainsi, l'évaluateur peut procéder à l'évaluation de l'utilité et de l'utilisabilité de l'interface en une seule étape tout en gardant un comportement normal du composant par rapport au concepteur (aucune lenteur d'exécution ou bien spécification d'évaluation ne doit être apparente pour l'utilisateur de ces composants.). De plus, il serait plus intéressant de concevoir un rapport commun qui englobe les problèmes issus de l'inspection d'utilisabilité et le test utilisateur de l'interface graphique. Notons que le rapport proposé concernant l'inspection de l'utilité par cette version de composants reste assez difficile à exploiter par l'évaluateur. En effet, l'évaluateur dispose d'un ensemble de données numériques pour l'évaluation. Comme nous l'avons spécifié dans la section précédente, le mouchard proposé ne permet pas d'exploiter et d'enregistrer les données recueillies pour mieux les exploiter. Une idée consiste à exploiter les services web pour l'évaluation. Ces services assurent l'inspection de l'utilisabilité. Ainsi, le rôle de composants dédiés pour l'inspection de l'utilisabilité sera restreint à communiquer ces attributs comme étant une entée à ces web service et par conséquent ces services engloberont les différents mécanismes d'évaluation.

Une autre perspective consiste à doter les composants proposés par une sorte d'intelligence. En effet ; cette perspective vise à exploiter les travaux issus du domaine de l'intelligence artificielle et plus précisément de la thématique des systèmes multi-agents. Les composants pourraient communiquer entre eux et trouver la meilleure présentation possible pour la présentation de l'interface. Autrement dit, les composants composent eux même l'interface graphique grâce à un ensemble de logiques et de règles issues de l'ergonomie logicielle. Le composant serait aussi doté de mécanismes permettant de se modifier pour être cohérent aux différentes règles ergonomiques.

### Autres perspectives

L'inspection de la qualité ergonomique des interfaces Homme-Machine est une technique de plus en plus utilisée pour l'évaluation des *IHM*. Elle se base essentiellement sur des règles ergonomiques issues de l'ergonomie logicielle et d'autres domaines adjacents à l'ergonomie logicielle (Exemple : physiologie du travail, la sociologie, la biomécanique, etc.). Néanmoins, une réflexion se pose concernant ces règles. Cette dernières est utilisée comme étant une référence. La réflexion consiste à étudier la validité des règles utilisées. En effet, nous nous proposons d'évaluer les règles elles mêmes avant de les exploiter pour l'évaluation des interfaces graphiques. Cette réflexion est alimentée par le fait que la majorité des règles datent de plus que cinq ans. Notons que l'utilisateur des systèmes interactifs subit des changements important par rapport à son niveau d'expertise et son expérience avec les grandes avancées de la technologie et d'internet. Effectivement, l'utilisateur dispose de plus en plus d'accès à différents systèmes interactifs et aux sites web. Ainsi, son expérience et sa vision des systèmes interactifs l'utilisateur connait tous les jours de plus en plus d'expérience.

La deuxième perspective porte sur l'évaluation des applications de type trois dimensions. Leur usage est de plus en plus fréquent (jeux vidéos, simulation, etc.). Néanmoins, les travaux portant sur l'évaluation de ce type d'application reste assez rares. Il serait ainsi intéressent de procéder à concevoir une méthodologie pour l'évaluation de l'interaction entre l'utilisateur et ces interfaces.

La troisième ontologie concerne l'exploitation des ontologies pour l'évaluation des *IU* . L'idée est de générer des résultats (nouvelles connaissances) d'une évaluation en se basant sur une ontologie (de domaine)(de Oliveira *et al.*, 2013). La première étape consiste à préparer la base de connaissance en commençant par le développement de l'ontologie (taxonomie, classes, relations, etc.). L'étape suivante consiste à instancier l'ontologie par des données réelles (population de l'ontologie). Ensuite, l'inférence de l'ontologie permet de générer des nouvelles connaissances en exploitant la taxonomie (hiérarchie

des classes), les relations et les axiomes (rules). Ces nouvelles connaissances ne sont que les résultats qui devraient être fournis à l'utilisateur. Dans le cas de l'évaluation des *IU*, les règles de l'évaluation sont définies par les relations entres concepts et les axiomes pour avoir plus de connaissances.

# Quatrième partie

# Bibliographie

# Bibliographie


A-Prompt : A-prompt. http ://www.aprompt.snow.utoronto.ca/, October 2011.

J. Abascal, M. Arrue, I. Fajardo, N. Garay et J. Toms : The use of guidelines to automatically verify web accessibility. *Univers. Access Inf. Soc.*, 3(1):71–79, mars 2004. ISSN 1615-5289. URL http://dx.doi.org/10.1007/s10209-003-0069-3.

M. Abed : *Méthodes et Modèles formels et semi-formels de conception et évaluation des systèmes homme-machine.* Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, France, 2001.

G. Abowd, R. Beale, A. Dix et J. Finlay : *Human-Computer Interaction.* ISBN : 0130461091. New York : Prentice Hall, 2004.

A. Abran, A. Khelifi, W. Suryn et A. Seffah : Usability meanings and interpretations in iso standards. *Control, Software Quality*, 11(4):325–338, november 2003. ISSN 0963-9314. URL http://dx.doi.org/10.1023/A:1025869312943.

W.I. Act : Section 508 standards. http ://www.section508.gov/, September 2012.

A. Almeida et C. Baranauskas : Universal design principles combined with web accessibility guidelines a case study. *In Proceedings of the IX Symposium on Human Factors in Computing Systems*, IHC '10, pages 169–178, Porto Alegre, Brazil, Brazil, 2010. Brazilian Computer Society. URL http://dl.acm.org/citation.cfm?id=1999593.1999612.

S. Alter : Designing and engineering for emergence : A challenge for hci practice and research. *AIS Transactions on Human-Computer Interaction*, 2(4):127–140, 2010.

B. Aman : *Encyclopedia of Database Systems*, chapitre XSL/XSLT, pages 3676–3681. Springer, 2009.

J.A. Anderson et T. Head : *Automata theory with modern applications.* Numéro ISBN 978-0-521-84887-9. Cambridge University Press, 2006.

ANDROID : User interface guideline. www.developer.android.com, September 2012.

M. ARRUE, M. VIGO et J. ABASCAL : Automatic evaluation of mobile web accessibility. *In* Constantine STEPHANIDIS et Michael PIEPER, éditeurs : *Universal Access in Ambient Intelligence Environments*, volume 4397 de *Lecture Notes in Computer Science*, pages 244–260. Springer Berlin Heidelberg, 2007a. ISBN 978-3-540-71024-0. URL http://dx.doi.org/10.1007/978-3-540-71025-7-16.

M. ARRUE, M. VIGO, A. AIZPURUA et J. ABASCAL : Accessibility guidelines management framework. *In* Constantine STEPHANIDIS, éditeur : *Universal Access in Human-Computer Interaction. Applications and Services*, volume 4556 de *Lecture Notes in Computer Science*, pages 3–10. Springer Berlin / Heidelberg, 2007b. ISBN 978-3-540-73282-2. 10.1007/978-3-540-73283-9-1.

ATRC : A-prompt : Web accessibility verifier. Rapport technique, Center, Adaptive Technology Resource and Trace Center, Canada & USA, 2002.

C. BACH et D.L. SCAPIN : Critères ergonomiques pour les interactions homme-environnements virtuels définitions, justifications et exemples. Rapport technique, INRIA, N 5531, 2005.

C. BACH et D.L. SCAPIN : Comparing inspections and user testing for the evaluation of virtual environments. *International Journal of Human-Computer Interaction*, 26(8):786–824, 2010. URL http://www.tandfonline.com/doi/abs/10.1080/10447318.2010.487195.

R. BAGGEN, J. CORREIA, K. SCHILL et J. VISSER : Standardized code quality benchmarking for improving logiciel maintainability. *Logiciel Quality Journal*, pages 1–21, 2011.

S. BALBO : Ema : Automatic analysis mechanism for the ergonomic evaluation of user interfaces. Rapport technique, DSIRO Division of Information Report 96/44, 1996.

I. BARK et A.F. GULLIKSEN : Use and usefulness of hci methods : Results from an exploratory study among nordic hci practitioners. *Poeple and computer XIX - The bigger picture*, 2:201–217, 2006.

L. BARKHUUS et J. RODE : From mice to men. *In 24 Years of Evaluation in CHI*, 2007. URL http://www.viktoria.se/altchi/.

J. M. C. BASTIEN et D. L. SCAPIN : Critères ergonomiques pour l'évaluation d'interfaces utilisateurs. Rapport technique, INRIA : Le Chesnay, N 156, 1993.

J. M. C. Bastien et D. L. Scapin : Evaluation des systèmes d'information et critères ergonomiques. *In* C. Kolski, éditeur : *Environnement évolués et évaluation de l'IHM. Interaction Homme Machine pour les SI*, volume 2, pages 53–80. Hermes, 2001.

J. M. C. Bastien et D. L. Scapin : Les méthodes ergonomiques : de l'analyse à la conception et à l'évaluation. *In Ergonomie et Informatique Avancée, Ergo-IA'2002, A. Drouin, J. Robert (Eds), I.D.L.S., Biarritz, 8-10 octobre*, 2002.

T. Beckers et N. Fuhr : User-oriented and eye tracking-based evaluation of an interactive search system. *In 4th Workshop on Human-Computer Interaction and Information Retrieval*, 2010.

A. Beirekdar : *A methodology for automating web usability and accessibility evaluation by guideline*. Thèse de doctorat, Facultés Universitaires Notre-Dame de la Paix, Namur, 2004.

A. Beirekdar, J. Vanderdonckt et M. Noirhomme-fraiture : A framework and a language for usability automatic evaluation of web sites by static analysis of html source code. *In Proceedings of 4 th International Conference on Computer-Aided Design of User Interfaces*, pages 15–17. Kluwer Academics Pub, 2002.

B. Beizer : *Techniques, Software Testing*. London : International Thompson Computer Press, 1990.

R. Bentley, J. A. Hughes, D. Randall, T. Rodden, P. Sawyer, D. Shapiro et I. Sommerville : Ethnographicallyinformed systems design for air traffic control. *In the Conference on Computer Supported Cooperative Work. (Toronto, November), 123-129*, 1992.

N. Bevan : Quality in use : Meeting user needs for quality. *Journal of Systems and Software*, 49(1):89 – 96, 1999. ISSN 0164-1212. URL http://www.sciencedirect.com/science/article/pii/S0164121299000709.

H. Beyer et K. Holtzblatt : *Contextual design : Defining Customer-Centered Systems*, volume 6. Sans Francisco : Morgan Kaufmann Publishers, New York, USA, janvier 1998. URL http://doi.acm.org/10.1145/291224.291229.

M. Billi, L. Burzagli, T. Catarci, G. Santucci, E. Bertini, F. Gabbanini et E. Palchetti : A unified methodology for the evaluation of accessibility and usability of mobile applications. *Univers. Access Inf. Soc.*, 9(4):337–356, novembre 2010. ISSN 1615-5289. URL http://dx.doi.org/10.1007/s10209-009-0180-1.

A.E. Blandford, J.K. Hyde, T. R. G. Green et I. Connell : Scoping analytical usability evaluation methods : A case study. *Human-Computer Interaction*, 23(3):278–327, 2008. URL http://www.tandfonline.com/doi/abs/10.1080/07370020802278254.

T. Bonte, H. Ezzedine, S. Hammadi, S. Hayat, C. Kolski, M. Ould Sidi, C. Tahon, A. Trabelsi et F. Malca : Rapport final du projet sart : Système d'aide à la régulation de trafic du réseau de transport valenciennois et de ses pôles d'échanges, décembre 2007. Rapport technique, SART, 2007.

N. Bowers : Weblint : quality assurance for the world wide web. *Comput. Netw. ISDN Syst.*, 28(7-11):1283–1290, mai 1996. ISSN 0169-7552. URL http://dx.doi.org/10.1016/0169-7552(96)00038-4.

E. Brangier : *Besoin et interface.* In J., I. Akoka., Comyn-Wattiau. Encyclopédie de l'informatique et des systèmes d'information. 1070-1084, Paris : Vuibert, 2007.

T. Brinck et E. Hofer : Automatically evaluating the usability of web sites. *In CHI '02 extended abstracts on Human factors in computing systems*, CHI EA '02, pages 906–907, New York, USA, 2002. ACM Press. ISBN 1-58113-454-1. URL http://doi.acm.org/10.1145/506443.506652.

J. Brooke : Sus : A "quick and dirty" usability scale. *In In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.) Usability Evaluation in Industry (pp. 189-194). London : Taylor and Francis*, 1996.

B. Buxton : *Sketching User Experiences : Getting the Design Right and the Right Design (Interactive Technologies).* Morgan Kaufmann, 1 édition, March 2007.

J. M. Carroll : *MacMillan Encyclopedia of Cognitive Science.* "MacMillan Encyclopedia of Cognitive Science". London : Macmillan-Nature Publishing Group, 2002.

T. Carta, F. Paterno et V. Santana : Support for remote usability evaluation of web mobile applications. *In SIGDOC'11, Pisa, Italy, October 3Ű5, 129-136*, 2011.

A.S. Carter et C.D. Hundhausen : How is user interface prototyping really done in practice ? a survey of user interface designers. *In Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, VLHCC '10, pages 207–211, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4206-5. URL http://dx.doi.org/10.1109/VLHCC.2010.36.

C.E.Analytics : Clicktale. http ://www.clicktale.com, September 2012.

V. L. Centeno, C. D. Kloos, B. C. del Toro et M. Gaedke : Web accessibility evaluation via xslt. *In* Mathias Weske, Mohand-Saad Hacid et Claude Godart, éditeurs : *Web Information Systems Engineering WISE 2007 Workshops*, volume 4832 de *Lecture Notes in Computer Science*, pages 459–469. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-77009-1. URL http://dx.doi.org/10.1007/978-3-540-77010-7-45. 10.1007/978-3-540-77010-7-45.

V. L. Centeno, C. D. Kloos, M. Gaedke et M. Nussbaumer : Waex : Web accessibility evaluator in a single xslt file. *Automated Specification and Verification of Web Systems*, pages 29–36, 2006.

R. Chalon : *Réalité Mixte et Travail Colla boratif : IRVO, un modèle de l'Interaction Homme-Machine*. Thèse de doctorat, Ecole centrale de Lyon, 2004.

S. Charfi, E. Dubois et D.L. Scapin : Usability recommendations in the design of mixed interactive systems. *In Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '09, pages 231–236, New York,USA, 2009. ACM. ISBN 978-1-60558-600-7. URL http://doi.acm.org/10.1145/1570433.1570475.

S. Charfi, H. Ezzedine, C. Kolski et F. Moussa : Towards an automatic analysis of interaction data for hci evaluation application to a transport network supervision system. *In* Julie Jacko, éditeur : *Human-Computer Interaction. Design and Development Approaches*, volume 6761 de *Lecture Notes in Computer Science*, pages 175–184. Springer Berlin / Heidelberg, 2011a. ISBN 978-3-642-21601-5. URL http://dx.doi.org/10.1007/978-3-642-21602-2-20. 10.1007/978-3-642-21602-2-20.

S. Charfi, A. Trabelsi, H. Ezzedine et C. Kolski : Graphical controls based environment for user interface evaluation. *In* Marco Winckler, Peter Forbrig et Regina Bernhaupt, éditeurs : *Human-Centered Software Engineering*, volume 7623 de *Lecture Notes in Computer Science*, pages 360–367. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34346-9. URL http://dx.doi.org/10.1007/978-3-642-34347-6-28. 10.1007/978-3-642-34347-6-28.

S. Charfi, A. Trablesi, H. Ezzedine et Kolski C : Towards ergonomic guidelines integration within graphical interface controls for the evaluation of the is. *In MSLT 2011, First IEEE International Conference on Mobility, Security and Logistics in Transport (May 31 - June 1-3), Hammamet, Tunisia, 76-82.*, 2011b.

M. Chen, J. R. Anderson et M. Sohn : What can a mouse cursor tell us more ? correlation of eye/mouse movements on web browsing. *In Ext. Abstracts CHI 2001, ACM Press*, 2001.

Clixpy : Clixpy. http ://www.clixpy.com/, October 2007.

G. Cockton, A. Woolrych et M. Hindmarch : Reconditioned merchandise : Extended structured report formats in usability inspection. *In CHI 2004 Extended Abstracts, 1433-36. ACM Press : New York*, 2004.

M. Cooper : *Evaluating Accessibility and Usability of Web Pages.* Computer-Aided Design of User Interfaces II, Louvain-la-Neuve, 1999.

L. Copeland : *A Practitioner's Guide to Software Test Design.* Boston : Artech House Publishers, 2004.

J. Coutaz : *Interface homme-ordinateur, conception et réalisation.* Dunod, Paris, 1990.

B. Crandall, G. Klein et R. R. Hoffman : *Working Minds : A Practitioner's Guide to Cognitive Task Analysis.* MIT Press, 2006.

CrazyEgg : Crazyegg. http ://www.crazyegg.com, September 2012.

A. Crystal et B. Ellington : Task analysis and human-computer interaction : approaches, techniques, and levels of analysis. *In the Tenth Americas Conference on Information Systems, New York, August 2004*, 2004.

M.M. Danchak : Crt displays for power plants. *Instrumentation Technology*, 23(10):29–36, 1976.

KÃąthia MarÃğal de Oliveira, Firas Bacha, Houda Mnasser et Mourad Abed : Transportation ontology definition and application for the content personalization of user interfaces. *Expert Systems with Applications*, 40(8):3145 – 3159, 2013. ISSN 0957-4174. URL http://www.sciencedirect.com/science/article/pii/S0957417412012699.

G. Delannay : A generic traceability tool. http ://www.info.fundp.ac.be/ pth/fundp-docs/gde.pdf, 2003.

Delmia : Delmia quest. www.cardix.com/pdf/delmia/, October 2012.

A. Dillon et M. Morris : User acceptance of new information technology : theories and models. *Williams Annual Review of Information Science and Technology Medford NJ : Information Today*, 31:3–2, 1996.

A. Dix : Human computer interaction in the early 21st century : a stable discipline, a nascent science, and the growth of the long tail. *In SIGCHI Ireland Inaugural Lecture, Trinity College, Dublin.*, 2008.

A. Dixon, J. Finaly, G. D. Abows et R. Beale : *Human-Computer Interaction*. Education, Pearson, 3rd edition édition, 2001.

S. Dray et D. Siegel : Remote possibilities ? : international usability testing at a distance. *interactions*, 11(2):10–17, mars 2004. ISSN 1072-5520. URL http://doi.acm.org/10.1145/971258.971264.

J.F. Dumas et J.C. Redish : *A Practical Guide to Usability Testing*. Greenwood Publishing Group Inc., Westport, CT, USA, 1993. ISBN 089391990X.

M. Etgen et J. Cantor : What does getting wet (web event-logging tool) mean for web usability ? *In Fifth Human Factors and the Web Conference*, 1999.

H. Ezzedine, T. Bonte, C. Kolski et C. Tahon : Intermodal transportation system management : towards integration of traffic management system and users information system. *In Fuchun Sun, Huaping Liu (Ed.), IMACS Multiconference on Computational Engineering in Systems Applications (CESA'2006, Beijing, China, oct. 4-6, 2006), volume 1, Tsinghua University Press, page 972-979, octobre, ISBN 7-302-13922-9.*, 2006.

H. Ezzedine, T. Bonte, C. Kolski et C. Tahon : Integration of traffic management and traveller information systems : basic principles and case study in intermodal transport system management. *International Journal of Computers, Communications & Control (IJCCC)*, 3:281–294, 2008.

D. Fallman et J. A. . Waterworth : Dealing with user experience and affective evaluation in hci design : A repertory grid approach. *In Workshop Paper, CHI 2005, Conference on Human Factors in Computing Systems, April 2-7, Portland, Oregon, ACM Press*, 2005.

P. Faraday : Visually critiquing web pages. *In 6th Conf. on Human Factors & the Web, Austin, Texas,*, 2000. URL http://facweb.cs.depaul.edu/cmiller/faraday/Faraday.HTML.

C. Farenc : *ERGOVAL : une méthode de structuration des règles ergonomiques permettant l'évaluation automatique d'interfaces graphiques*. Thèse de doctorat, Université Toulouse 1, 1997.

C. Farenc, P. A. Palanque, J. M.C. Bastien, D. L. Scapin et M. Winckler : Towards a general guidance and support tool for usability optimization. *In* C. Stephanidis, éditeur : *HCI*, pages 160–164. Lawrence Erlbaum, 2001. ISBN 0-8058-3609-8. URL http://dblp.uni-trier.de/db/conf/hci/hci2001-3.html#FarencPBSW01.

T. W. Fiora, S. Baker, I. Warren et G. Dobbie : Automated usability testing framework. *In Proceedings of the ninth conference on Australasian user interface - Volume 76*, AUIC '08, pages 55–64, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc. ISBN 978-1-920682-57-6. URL http://dl.acm.org/citation.cfm?id=1378337.1378349.

R. Fitzpatrick : The software quality star : A conceptual model for the software quality curriculum. *In Workshop paper for Closing the Gaps : Software Engineering and Human-Computer Interaction at INTERACT 2003 : Ninth IFIP TC 13 International Conference on Human-Computer Interaction, September 2003, Zurich, Switzerland*, 2003.

A. P. Freire et R.P. de Mattos Fortes : Automatic accessibility evaluation of dynamic web pages generated through xslt. *In Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility*, W4A '05, pages 81–84, New York, USA, 2005. ACM. ISBN 1-59593-219-4. URL http://doi.acm.org/10.1145/1061811.1061826.

Y. Gabillon, G. Calvary et H. Fiorino : Composition d'interfaces homme-machine en contexte : approche par planification automatique. *Technique et Science Informatiques (TSI)*, 30(10):1143–1166, 2011. Numéro Spécial Composition, Décembre 2011.

W.O. Galitz : *The Essential Guide to User Interface Design : An Introduction to GUI Design Principles and Techniques*. Numéro ISBN-10 : 0470053429. Wiley, 2007.

S.C. Gibbons : Toward an expert system based menu interface evaluation tool. Rapport technique, INRIA, N 1581, 1992.

Google : Google analytics. http ://www.google.com/analytics/, October 2011.

D. Grammenos, D. Akoumianakis et C. Stephanidis : Integrated support for working with guidelines : the sherlock guideline management system. *Interacting with Computers*, 12(3):281 – 311, 2000. ISSN 0953-5438. URL http://www.sciencedirect.com/science/article/pii/S0953543899000156.

S. Greenberg et B. Buxton : Usability evaluation considered harmful (some of the time). *In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 111–120, New York, USA, 2008. ACM Press. ISBN 978-1-60558-011-1. URL http://doi.acm.org/10.1145/1357054.1357074.

M. Grislin et C. Kolski : Evaluation des interfaces homme-machine lors du développement de système interactif. *Technique et Science Informatiques (TSI)*, 3:265–296, 1996.

Nielsen Norman Group : Lift for dreamweaver. http ://www.usablenet.com/ products services/ lfdnng/ lfdnng.HTML, December 2011.

Open Hallway : http ://www.openhallway.com/, September 2012.

H.R. Hartson et D. Hix : Towards empirically derived methodologies and tools for human-computer development. *International Journal of Man-Machine Studies*, 31: 477–494, 1989.

S. R. Haynes, S. Purao et A. L. Skattebo : Situating evaluation in scenarios of use. *In In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (Chicago, Illinois, USA, November 06 - 10, 2004). CSCW '04. ACM, New York, 2004, 92-101*, 2004.

M. A. Hearst : *Search User Interface*. Press, Cambridge Univerty, 2009.

T.T. Hewett : Acm sigchi curricula for human-computer interaction. Rapport technique, Drexel Univ., Philadelphia, PA, New York, U. S. A., 1992.

HIIS : Concur task trees environment. http ://giove.isti.cnr.it/ctte.html, September 2012.

D.M. Hilbert et D.F. Redmiles : Extracting usability information from user interface events. *ACM Comput. Surv.*, 32(4):384–421, décembre 2000. ISSN 0360-0300. URL http://doi.acm.org/10.1145/371578.371593.

A. Holzinger : Rapid prototyping for a virtual medical campus interface. *IEEE Softw.*, 21(1):92–99, janvier 2004. ISSN 0740-7459. URL http://dx.doi.org/10.1109/MS.2004.1259241.

J.A. Hong, J. Heer, S. Waterson et J.A. Landay : Webquilt : A proxy-based approach to remote web usability testing. *In ACM Transactions on Information Systems, Vol. 19, No. 3, July 2001*, pages 263–285, 2001. URL http://doi.acm.org/10.1145/1570433.1570475.

S. Howard et D.M. Murray : A taxonomy of evaluation techniques for hci. *In* H.-J. Bullinger et B. Shakel, éditeurs : *Human-Computer Interaction : INTERACT'87*, pages 453–459. North-Holland, Amsterdam, 1987.

E. Hvannberg, E. Law et M. Lárusdóttir : Heuristic evaluation : Comparing ways of finding and reporting usability problems. *Interacting with Computers*, 19(2):225–240, 2007.

W. Hwang et G. Salvendy : Number of people required for usability evaluation : the 10 2 rule. *Commun. ACM*, 53(5):130–133, mai 2010. ISSN 0001-0782. URL http://doi.acm.org/10.1145/1735223.1735255.

Insydelabs : Evalyzer. http ://www.evalyzer.com/, Mars 2013.

1998 ISO : Iso 9241-12 :1998 ergonomic requirements for office work with visual display terminals (vdts) – part 12 : Presentation of information. Rapport technique, International Organization for Standardization, 1998.

1999 ISO : Iso 13407 :1999, human-centered design processes for interactive systems. Rapport technique, International Organization for Standardization, 1999.

M.Y. Ivory et M.A. Hearst : The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, décembre 2001. ISSN 0360-0300. URL http://doi.acm.org/10.1145/503112.503114.

R. Jaaskelainen : *Focus on Methodolody in think-aloud studies on translating.* Tirkkonen-Condit, Sonja/Jääskeläinen, Rita (eds.), Tapping and mapping the processes of translation and interpreting. Outlooks on empirical research. Amsterdam/-Philadelphia : Benjamins, 71-82, 2000.

E. Jackson-Sanborn, K. Odess-Harnish et N. Warren : Web site accessibility : a study of six genres. *Library Hi Tech*, 20(3):308–317, 2002.

N. E. Jacobsen, M. Hertzum et B. E. John : The evaluator effect in usability tests. *In CHI 1998, ACM Press*, 1998.

C. Jarrett et G. Gaffney : *Forms that work : Designing Web forms for usability.* San Francisco : Morgan Kaufmann/Elsevier, 2009.

R. Jeffries, J. R. Miller, C. Wharton et K. Uyeda : User interface evaluation in the real world : a comparison of four techniques. *In Proceedings of the SIGCHI conference on Human factors in computing systems : Reaching through technology*, CHI '91, pages 119–124, New York, USA, 1991. ACM Press. ISBN 0-89791-383-3. URL http://doi.acm.org/10.1145/108844.108862.

J. Jiang, E. Murphy, S. Bailin, W. Truszkowski et M. Szczur : Prototyping a knowledge based compliance checker for ui evaluation in motif development environments. *In In MOTIF'92 : Second Annual International Motif Users Meeting Bethesda, MD, Washington, DC : Open Systems, page 258-268*, 1992.

J. Johnson : *Gui Bloopers : Don'ts and Do's for Software Developers and Web Desigers.* Morgan Kaufmann, 1 édition, 2000.

M. Jovic, A. Adamoli, D. Zaparanuks et M. Hauswirth : Automating performance testing of interactive java applications. *In Proceedings of the 5th Workshop on Automation of Software Test, ISBN 978-1-60558-970-1, page 8-15, Cape Town, South Africa, doi = http ://doi.acm.org/10.1145/1808266.1808268, ACM*, 2010.

M. Jovic et M. Hauswirth : Performance testing of gui applications. *In TESTBEDS '10 : Second International Workshop on TESTing Techniques & Experimentation Benchmarks for Event-Driven Software, Paris, France*, 2010.

M. Kapsi, E. Vlachogiannis, J. Darzentas et T. Spyrou : The usability of web accessibility guidelines : An approach for evaluation. *In Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Part III : Applications and Services*, UAHCI '09, pages 716–724, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02712-3. URL http://dx.doi.org/10.1007/978-3-642-02713-0-76.

E. Karapanos : Acceptability of nomadic user interfaces : An investigation upon user perceptions. Mémoire de D.E.A., University College London, 2005.

J. Karat et C. M. Karat : The evolution of user-centered focus in the human-computer interaction field. *IBM Syst. J.*, 42(4):532–541, octobre 2003. ISSN 0018-8670. URL http://dx.doi.org/10.1147/sj.424.0532.

N.V. Keith : Using guidelines to assist in the visualisation design process. *In proceedings of the 2005 Asia-Pacific symposium on Information visualisation, 45*, APVis '05, pages 115–123. Australian Computer Society, Inc., 2005. ISBN 1-920-68227-9. URL http://dl.acm.org/citation.cfm?id=1082315.1082332.

W. Kim et J. Foley Don : User interface presentation design assistant. *In in Proceedings of UIST'90, Snowbird, 3-5 October, page 10-20.*, 1990.

C. Kolski : *Analyse et conception de l'IHM, Interaction homme-machine pour les SI (tome 1) ; Environnements évolués et évaluation de l'IHM, Interaction homme machine pour les SI (tome 2).* Editions Hermes, Paris, 2001.

C. Kolski et H. Ezzedine : Conception et évaluation des ihm de supervision : éléments méthodologiques. *Génie Logiciel*, 65:2–11, 2003.

C. Kolski et P. Millot : A rule-based approach to the ergonomic "statique" evaluation of man-machine graphic interface in industrial processes. *International Journal of Man-Machine Studies*, 35:657–657, 1991.

J.A.. Konstan : Tutorial / hci for recommender systems : a tutorial. *In Proceedings of the 16th international conference on Intelligent user interfaces, ISBN 978-1-4503-0419-1, Palo Alto, CA, USA*, 2011.

P. Kortum : *HCI Beyond the GUI : Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces (Interactive Technologies)*. Morgan Kaufmann, 1 édition, 2008.

E. L. Koua et M.J. Kraak : A usability framework for the design and evaluation of an exploratory geovisualization environment. *In Information Visualisation (IV), 14-16 July, at London, UK*, 2004.

B. Kovacs, F. Gaunet et X. Briffault : *Les techniques d'analyse de l'activité pour l'IHM*. Hermes, 2004. ISBN 2-7462-0944-6.

O. Kummer, F. Wienberg, M. Duvigneau, M. Köhler, D. Moldt et H. Rölke : Renew - the reference net workshop. *In In Eric Veerbeek, editor, Tool Demonstrations. 24th International Conference on Application and Theory of Petri Nets (ATPN 2003). International Conference on Business Process Management (BPM 2003), 99-102*, 2003.

H. Kuusela et P. Paul : A comparison of concurrent and retrospective verbal protocol analysis. *American Journal of Psychology*, 113(3):387–404, 2000.

G. Kwon, D.H. Ham et W.C. Yoon : Evaluation of software usability using scenarios organized by abstraction structure. *In Proceedings of the 14th European conference on Cognitive ergonomics : invent ! explore !*, ECCE '07, pages 19–22, New York, USA, 2007. ACM. ISBN 978-1-84799-849-1. URL http://doi.acm.org/10.1145/1362550.1362557.

Labsmedia : Click heat. http ://www.labsmedia.com/clickheat/index.html, September 2012.

C. Larman et V. R. Basili : Iterative and incremental development : A brief history. *IEEE Computer*, 36(6):47–56, 2003.

D. Lavery, G. Cockton et M.P. Atkinson : Comparison of evaluation methods using structured usability problem reports. *Behaviour and Information Technology*, 16(4):246–266, 1997.

E. Law, E. Hvannberg et G. Cockton : *Maturing Usability Quality in Software*. Human-Computer Interaction Series., 1 édition, 2008.

E.L. Law et P. van Schaik : Editorial : Modelling user experience - an agenda for research and practice. *Interacting with Computers*, 22(5):131–322, 2010.

J. Lazar : *Universal Usability : Designing Computer Interfaces for Diverse User Populations*. John Wiley & Sons, 2007. ISBN 0470027274.

A. Lecerof et F. Paterno : Automatic support for usability evaluation. *IEEE Trans. on Software Engineering*, 24(10):863 – 888, 1998.

B. Leporini et F. Paterno : Increasing usability when interacting through screen readers. *Univers. Access Inf. Soc.*, 3(1):57–70, mars 2004. ISSN 1615-5289. URL http://dx.doi.org/10.1007/s10209-003-0076-4.

B. Leporini, F. Paterno et A. Scorcia : Flexible tool support for accessibility evaluation. *Interact. Comput.*, 18(5):869–890, septembre 2006. ISSN 0953-5438. URL http://dx.doi.org/10.1016/j.intcom.2006.03.001.

C. Lewis, P. Polson, C. Wharton et J. Rieman : Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *In Chi '90 Proceedings, ACM Press, pages 235-242*, 1990.

Loop11 : Remote & online usability test. http ://www.loop11.com/, September 2012.

J. Lowgren et T. Nordqvist : Knowledge-based evaluation as design support for graphical user interfaces. *In Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, pages 181–188, New York, USA, 1992. ACM Press. ISBN 0-89791-513-5. URL http://doi.acm.org/10.1145/142750.142786.

Clearleft Ltd : Silverback. http ://silverbackapagescom/, September 2012.

R. Mahajan et B. Shneiderman : Visual and textual consistency checking tools for graphical user interfaces. *IEEE Transactions on Software Engineering*, 23(11):722–735, 1997. ISSN 0098-5589.

T. Mahatody, M. Sagar et C. Kolski : State of the art on the cognitive walkthrough method, its variants and evolutions. *International Journal of Human-Computer Interaction*, 26(8):741–785, 2010. URL http://www.tandfonline.com/doi/abs/10.1080/10447311003781409.

C. Mariage : *MetroWeb - logiciel de support à l'évaluation de la qualité ergonomique des sites web.* Thèse de doctorat, Université Catholique de Louvain, 2005.

C. Mariage et J. Vanderdonckt : Creating contextualised usability guides for web sites design and evaluation. *In* RobertJ.K. Jacob, Quentin Limbourg et Jean Vanderdonckt, éditeurs : *Computer-Aided Design of User Interfaces IV*, pages 147–158. Springer Netherlands, 2005. ISBN 978-1-4020-3145-8. URL http://dx.doi.org/10.1007/1-4020-3304-4-12.

C. Mariage, J. Vanderdonckt, A. Beirekdar et M. Noirhomme : Destine : outil d'aide à l'évaluation de l'ergonomie des sites web. *In Proceedings of the*

*16th conference on Association Francophone d'Interaction Homme-Machine*, IHM 2004, pages 117–124, New York, USA, 2004. ACM. ISBN 1-58113-926-8. URL http://doi.acm.org/10.1145/1148613.1148630.

A. Mattelart : *Histoire de la société de l'information*. Numéro 978-2707148667. Editions La Découverte, 3 édition, March 2006.

D.W. Mauney et C. Masterton : Small screen interfaces. *In in Beyond The GUI edited by Philip Kortum*, 2008.

D. Mayhew : *The usability engineering lifecycle*. Morgan Kaufmann Publishers, 1999.

N. McNamara et J. Kirakowski : Measuring user-satisfaction with electronic consumer products : The consumer products questionnaire. *International Journal of Human-Computer Studies*, 69(6):375 – 386, 2011. ISSN 1071-5819. URL http://www.sciencedirect.com/science/article/pii/S1071581911000073.

N. Medvidovic et V. Jakobac : Using software evolution to focus architectural recovery. *Automated Software Engineering*, 13:225–256, 2005.

B. Meyer : *Object-Oriented Software Construction*. Hall, Prentice and River, Upper Saddle and Jersey, New, 2 édition, 1997.

G. Michel, W. De Abreu et E. .. Brangier : Electoral ergonomic guidelines to solve the interference of new technologies and the dangers of their broader use in computerised voting. *In ECGE'07 : 7th European Conference on E-Government. Den Haag, Netherlands, June 21-22, 337-348*, 2007.

Microsoft : Report definition language specification, third edition, 2010.

Microsoft : Windows user experience interaction guidelines. http ://download.microsoft.com/download/e/1/9/e191fd8c-bce8-4dba-a9d5-2d4e3f3ec1d3/UXGuide.pdf, December 2011. 2010.

F. Millerand et O. Martial : Guide pratique de conception et d'évaluation de sites web ( version 1). Rapport technique, Centre de recherché informatique de Montréal, 2001.

R. Molich, A.D. Thomsen, B. Karyukina, L. Schmidt, M. Ede, W. van Oel et M. Arcuri : Comparative evaluation of usability tests. *In CHI '99 extended abstracts on Human factors in computing systems*, CHI EA '99, pages 83–84, New York, USA, 1999. ACM Press. ISBN 1-58113-158-5. URL http://doi.acm.org/10.1145/632716.632768.

A.F. Monk, J. Carroll, S. Parker et M. Blythe : Why are mobile phones annoying ? *Behaviour and Information Technology*, 23:33–42, 2004.

M. Morandini, R. de Moraes, R. Leite, M.V. Cerrato et C.M. Lordello : Project and development of ergocoin version 2.0. *In Proceedings of the 14th international conference on Human-computer interaction : design and development approaches - Volume Part I*, HCII'11, pages 471–479, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21601-5. URL http://dl.acm.org/citation.cfm?id=2022384.2022440.

F. Moussa, C. Kolski et M.ă Riahi : Analyse des dysfonctionnements des systèmes complexes en amont de la conception des ihm : apports, difficultés, et étude de cas. *In Revue d'Interaction Homme Machine (RIHM), 7, 79-11*, 2006.

F. Mueller et A. Lockerd : Cheese : Tracking mouse movement activity on websites, a. tool for user modeling. *In Ext. Abstracts CHI 2001, ACM Press*, 2001.

D. Navarre : *Contribution à l'ingénierie en Interaction Homme-Machine. Une technique de description formelle et un environnement pour une modélisation et une exploitation synergiques des tâches et du système.* Thèse de doctorat, University of Toulouse 1, 2001.

U. Nerurkar : Web user interface design : Forgotten lessons. *IEEE Softw.*, 18(6):69–71, novembre 2001. ISSN 0740-7459. URL http://dx.doi.org/10.1109/52.965807.

NetMechanic : Html toolbox. www.netmechanic.com/products/HTML_Toolbox October 2011.

E. Newcomer et G. Lomow : *Understanding SOA with web services.* Addison-Wesley Professional, 1 édition, 2004. ISBN-13 : 978-0321180865.

J. Nielsen : *Engineering, Usability.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 0125184050.

J. Nielsen : The use and misuse of focus groups. *Software IEEE*, 14(1):94–95, 1997.

J. Nielsen et T. Landauer : A mathematical model of the finding of usability problems. *In Proceedings of ACM INTERCHI'93 Conference, Amsterdam, The Netherlands, 24-29 April, 206-213*, 1993.

J. Nielsen et R. Molich : Heuristic evaluation of user interfaces. *In Proceedings of the SIGCHI conference on Human factors in computing systems : Empowering people*, CHI '90, pages 249–256, New York, USA, 1990. ACM Press. ISBN 0-201-50932-6. URL http://doi.acm.org/10.1145/97243.97281.

J. Nielsen et M. Tahir : *Homepage Usability : 50 Websites Deconstructed*. New Riders Publishing, Thousand Oaks, CA, USA, 2001. ISBN 073571102X.

NIST : Nist web metrics. http ://zing.ncsl.nist.gov/WebTools/index.HTML, October 2011.

Ocawa : accessibility validator. www.ocawa.com/, October 2011.

E. L. Olmsted-Hawala, D. Elizabeth, S. Hawala et K. T. Ashenfelter : Think-aloud protocols : a comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. *In Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 2381–2390, New York, USA, 2010. ACM Press. ISBN 978-1-60558-929-9. URL http://doi.acm.org/10.1145/1753326.1753685.

H. Olsen : An evaluation of danish qualitative interview investigations. *Nordisk Psykologi*, 54(2):145–172, 2002.

Jr. Olsen et R. Dan : Evaluating user interface systems research. *In Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 251–258, New York, USA, 2007. ACM. ISBN 978-1-59593-679-0. URL http://doi.acm.org/10.1145/1294211.1294256.

R. Olsen et Jr. Dan : *User Interface Management Systems : Models and Algorithms*. Morgan Kaufmann Publishers, 1991.

C. Osterbauer, M. Lackner et G. Weinberger : Usability in chemical engineering. *Trends in Chemical Engineering*, 13:1–15, 2010.

L. Paganelli et F. Paterno : Intelligent analysis of user interactions with web applications. *In Proceedings of the 7th international conference on Intelligent user interfaces*, IUI '02, pages 111–118, New York, USA, 2002. ACM. ISBN 1-58113-459-2. URL http://doi.acm.org/10.1145/502716.502735.

L. Paganelli et F. Paterno : Tools for remote usability evaluation of web applications through browser logs and task models. *Behavior Research Methods, Instruments, and Computers*, 35(3):369–378, 2003.

P. Palanque et R. Bastide : Interactive cooperative objects : an object-oriented formalism based on petri nets for user interface design. *In Proceedings of the IEEE / System Man and Cybernetics 93 "Systems Engineering in the Service of Humans", Le Touquet, Octbre 1993, 17-20*, 1993.

B. Pan, H. Hembrooke, G. Gay, L. Granka, M. Feusner et J. Newman : The determinants of web page viewing behavior : An eye tracking study. *In ETRA 2004*, 2004.

F. Paterno : Concurtasktrees : An engineered notation for task models. *In The Handbook of Task Analysis for Human-Computer Interaction edited by Diaper, D., Stanton, N., 483-501.*, 2000.

F. Paterno et G. Ballardin : Remusine : a bridge between empirical and model-based evaluation when evaluators and users are distant. *Interacting with Computers*, pages 229–251, 2000.

F. Paterno, C. Mancini et S. Meniconi : Concurtasktrees : A diagrammatic notation for specifying task models. *In Proceedings Interact'97, 362-369, Sydney, Chapman&Hall*, 1997.

F. Paterno, A. Piruzza et C. Santoro : Remote web usability evaluation exploiting multimodal information on user behavior. *In* Gaelle Calvary, Costin Pribeanu, Giuseppe Santucci et Jean Vanderdonckt, éditeurs : *Computer-Aided Design of User Interfaces V*, pages 287–298. Springer Netherlands, 2007a. ISBN 978-1-4020-5820-2.

F. Paterno, A. Russino et C. Santoro : Remote evaluation of mobile applications. *In* Marco Winckler, Hilary Johnson et Philippe Palanque, éditeurs : *Task Models and Diagrams for User Interface Design 6th International Workshop, TAMODIA 2007, Lecture Notes in Computer Science, Vol. 4849, Winckler, M. ; Johnson, H. ; Palanque, P. (Eds.) ISBN : 978-3-540-77221-7, 155-169*, volume 4849 de *Lecture Notes in Computer Science*, pages 155–169. Springer Berlin / Heidelberg, 2007b. ISBN 978-3-540-77221-7.

P. Perrey et M. Lycett : Service-oriented architecture. *In Applications and the Internet Workshops, 27-31 January, 116-119*, 2003.

Bolt Peters : Ethnio. http ://ethn.io/, September 2012.

P.G. Polson, C. Lewis, J. Rieman et C. Wharton : Cognitive walkthroughs : a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36(5):741 – 773, 1992. ISSN 0020-7373. URL http://www.sciencedirect.com/science/article/pii/002073739290039N.

R. Pressman : *Logiciel Engineering : A Practitioner's Approach.* Boston, McGraw Hill, 2001.

M. Rauterberg : How to measure the ergonomic qualitiy of user interfaces in a task independent way. *Advances in Occupational Ergonomics and Safety*, 1:154–157, 1996.

Forrester Research : Annual report. Rapport technique, Forrester Research Inc, 1999. URL http://media.corporate-ir.net/mediafiles/irol/60/60569/reports/Forr99.pdf.

F. Ricca et P. Tonella : Web site analysis : Structure and evolution. *In Proceedings of the International Conference on Software Maintenance (ICSM'00)*, ICSM '00, pages 76–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0753-0. URL http://dl.acm.org/citation.cfm?id=850948.853419.

Y. Rogers, H. Sharp et J. Preece : *Interaction Design : Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, 2011.

J. Rubin et D. Chisnell : *Handbook of Usability Testing : How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, 2 édition, 2008. ISBN 0470185481, 9780470185483.

A. Rukshan et A. Baravalle : A quantitative approach to usability evaluation of web sites. *In Advances in Computing Technology, London, United Kingdom*, 2011.

G. Salvendy et L. Turley : Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, 21(2):137–143, 2002.

D.L. Scapin et J.M.C. Bastien : Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour and Information Technology,*, 6(4-5):220–231, 1997.

D.L. Scapin, C. Leulier, J. Vanderdonckt, C. Mariage, C. Bastien, C. Farescapinnc, P. Palanque et R. Bastide : A framework for organizing web usability guideline. *6th Conference on Human Factors & The web, Austin, Texas*, 2002.

E. Schaffer : *Institutionalization of Usability : A Step-by-Step Guide*. Numéro ISBN-13 : 978-0321179340. Addison-Wesley Professional, 2004.

M. Schmettow : Heterogeneity in the usability evaluation process. *In In : M. England, D. & Beale, R. (ed.), Proceedings of the HCI 2008, British Computing Society, 1, 89-98*, 2008.

M. Schrepp et P. Fischer : Goms models to evaluate the efficiency of keyboard navigation in web units. *Eminds-International Journal of Human Computer Interaction*, 1(2):33–46, 2007.

I. Scott MacKenzie : Fitts' law as a research and design tool in human computer interaction. *Human Computer Interaction*, 7:91–139, 1992. URL http://www.yorku.ca/mack/hci1992.pdf.

A. Seffah, M. Donyaee, R. B. Kline et H. K. Padda : Usability measurement and metrics : A consolidated model. *Software Quality Control*, 14(2):159–178, juin 2006. ISSN 0963-9314. URL http://dx.doi.org/10.1007/s11219-006-7600-8.

A. Seffah, H. Ezzedine, H. Javahery et S. Charfi : Reengineering and automating the user-oriented testing process. Submitted to Behaviour & Information Technology.

A. Seffah et H. Habieb-Mammar : Usability engineering laboratories : limitations and challenges toward a unifying tools/practices environment. *Behaviour & Information Technology*, 28(3):281–291, 2009. URL http://www.tandfonline.com/doi/abs/10.1080/01449290701803482.

A. Seffah et E. Metzker : *Adoption-centric Usability Engineering : Systematic Deployment, Assessment and Improvement of Usability Methods in Software Engineering*. Springer Publishing Company, Incorporated, 1 édition, 2008. ISBN 1848000189, 9781848000186.

B. Senach : Evaluation ergonomique des ihm : Une revue de la littérature. Rapport technique, INRIA N 1180, 1990.

A. Serna, H. Pigot, J. Bauchet, S. Giroux, V. Rialle, D. L-Desrochers et M. Mokhtari : Recommandations ergonomiques pour la conception de systèmes d'assistance cognitive dans les habitats intelligents. *In Proceedings of the Ergonomie et Informatique Avancee Conference*, Ergo'IA '10, pages 3–10, New York, USA, 2010. ACM. ISBN 978-1-4503-0273-9. URL http://doi.acm.org/10.1145/1868650.1868652.

B. Shneiderman : *Designing the user interface : Strategies for effective human-computer interaction*. Reading, MA : Addison-Wesley., 2 édition, 1992.

Feng Shui : Feng-gui. http ://www.feng-gui.com/, September 2012.

BCS SIGIST : Standard for software component testing (working draft 3.4). Rapport technique, British Computer Society Specialist Interest Group in Software Testing, 2001.

Speedzinemedia : Simple mouse tracking. http ://smt.speedzinemedia.com, September 2012.

J.C. Sperandio : *Traité d'ergonomie Cazamian, P., Hubault F. (eds.)*, chapitre L'apport de la psychologie du travail, pages 165–207. Octares, 1996.

StandardSBoard : *IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12*. Los Alamitos, CA, 1990.

C. Stephanidis : *Universal Access in HCI : Towards An information Society for All and (Human Factors and Ergonomics)*, volume 3(1). Press, C.R.C., 2001.

C. Stephanidis, D. Akoumianakis, J. Ziegler et K.P. Fähnrich : User interface accessibility : A retrospective of current standardisation efforts. *In* Gavriel Salvendy, Michael J. Smith et Richard J. Koubek, éditeurs : *HCI (1)*, pages 469–472. Elsevier, 1997. ISBN 0-444-82183-X. URL http://dblp.uni-trier.de/db/conf/hci/hci1997-1.html#StephanidisAZF97.

D. Stone, C. Jarrett, M. Woodroffe et S. Minocha : *User Interface Design and Evaluation (The Morgan Kaufmann Series in Interactive Technologies) (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN 0120884364.

Inc Sun Microsystems et Javasoft : *Java Look & Feel Design Guidelines*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st édition, 1999. ISBN 0201615851.

Y. Takata, T. Nakamura et H. Seki : Accessibility verification of www documents by an automatic guideline verification tool. *In Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4*, HICSS '04, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2056-1. URL http://dl.acm.org/citation.cfm?id=962752.962940.

J.C. Tarby : Evaluation précoce et conception orientée évaluation. *In 10ème conférence internationale Ergo'IA (Ergonomie et Informatique Avancée), Bidart/Biarritz, France, 11-13 Octobre, ISBN : 2-9514772-6-0*, 2006.

J.C. Tarby, H. Ezzedine et C. Kolski : Trace-based usability evaluation using aspect-oriented programming and agent-based software architecture. *In* A. Seffah, J. Vanderdonckt et M.C. Desmarais, éditeurs : *Human-Centered Software Engineering*, Human-Computer Interaction Series, pages 257–276. Springer London, 2009. ISBN 978-1-84800-906-6. URL http://dx.doi.org/10.1007/978-1-84800-907-3-12.

TAW : Servicios de accesibilidad y movilidad web. http ://www.tawdis.net/, October 2011.

TechSmith : Morae : Usability testing solution for web sites and software. http ://www.techsmith.com/Morae.asp., December 2012. 2007.

J. Thatcher, P.H. Burks, M.R.and Lauke, C. Heilmann, A. Kirkpatrick, B. Lawson, S. Henry, B. Regan, Rutter R. et M. Urban : *Web Accessibility : Web Standards And Regulatory Compliance*. Friends of Ed, 2006. ISBN-13 : 978- 1590596388.

Y.L. Theng et G. Marsden : Authoring tools : Towards continuous usability testing of web documents. *In In Proceedings of the 1st International Workshop on Hypermedia Development*, 1998.

M. Theofanos et W. Quesenbery : Towards the design of effective formative test reports. *Journal of Usability Studies*, 1:27–45, 2005.

M.A. Tinker : *Bases for Effective Reading*. University of Minnesota Press, Minneapolis, 1965.

M. Toomim, T. Kriplean, Claus Portner et J. Landay : Utility of human-computer interactions : toward a science of preference measurement. *In Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2275–2284, New York, USA, 2011. ACM Press. ISBN 978-1-4503-0228-9. URL http://doi.acm.org/10.1145/1978942.1979277.

A. Trabelsi : *Contribution à l'évaluation des systèmes interactifs orientés agents, application à un poste de supervision de transport urbain*. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis, 2006.

A. Trabelsi et H. Ezzedine : Evaluation of an information assistance system based on an agent-based architecture in transportation domain : first results. International Journal of Computers, Communications and Control, sous press, 2012.

A. Trabelsi, H. Ezzedine et C. Kolski : Architecture modelling and evaluation of agent-based interactive systems. *In SMC (6)'04*, pages 5159–5164, 2004.

C. Tran : *Vers un environnement générique et configurable pour l'aide à l'évaluation des systèmes interactifs à base d'agents. Application à un Système d'Aide à l'Information Voyageur*. Thèse de doctorat, University of Valenciennes and Hainaut Cambrésis, 2009.

C.D. Tran, H. Ezzedine et C. Kolski : Evaluation of agent-based interactive systems : Proposal of an electronic informer using petri nets. *Journal of Universal Computer Science*, pages 3202–3216, 2008.

D. Travis : 247 web usability guidelines. www.userfocus.co.uk/resources/guidelines.html, December 2011. 2009.

USABILLA : Usabilla. www.usabilla,com, September 2012.

USERFLY : Userfly. www.userfly.com, October 2009.

J. VANDERDONCKT : *Guide ergonomique des interfaces homme-machine*. Presse universitaire de Namur, Belgique, 1994.

J. VANDERDONCKT : Accessing guidelines information with sierra. *In Proc. of IFIP Interact'95*, pages 27–29. Chapman & Hall, 1995.

J. VANDERDONCKT : Development milestones towards a tool for working with guidelines. *Interacting with Computers*, 12(2):81 – 118, 1999. ISSN 0953-5438. URL http://www.sciencedirect.com/science/article/pii/S0953543899000193.

J. VANDERDONCKT et C. (Eds.) FARENC : *Tools for Working with Guidelines*. Springer, 2001.

R.A. VIRZI : Streamlining the design process : running fewer subjects. *In Proceeding of the Human Factors Scoiety. 34th Annual Meeting (Santa Monica, October 8/12) Orlando, The Human Factors and Ergonomics Society.*, 1990.

R.A. VIRZI : Refining the test phase of usability evaluation : how many subjects is enough ? *Hum. Factors*, 34(4):457–468, 1992. ISSN 0018-7208. URL http://dl.acm.org/citation.cfm?id=141691.141700.

W3C : W3c mobileok basic tests 1.0. http ://www.w3.org/TR/mobileOK-basic10-tests/, December 2008.

W3C : Evaluation and report language 1.0. http ://www.w3.org/TR/EARL10/, October 2011.

WATCHFIRE : Webxact. http ://webxact.watchfire.com/, October 2010.

S. WATERSON : Webquilt : A visu al analysis tool for understanding web usability clickstream data. Mémoire de D.E.A., Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, May 2002.

A. WATSON : Doctor watson, version 5.1. http ://watson.addy.com/, October 2011.

S. WATZMAN : *The Human Computer Interaction Handbook Sears A Jacko J.A.*, chapitre Visual design principles for usable interfaces, pages 263 – 285. 2002.

WCAG : Web content accessibility guidelines 1.0. http ://www.w3.org/TR/WCAG10/, october 2011.

A. WHITEFIELD, F. WILSON et J. DOWELL : A framework for human factors evaluation. *Behaviour & Information Technology*, 10:65–79, 1991.

M. A. Whiting, J. Haack et C. Varley : Creating realistic, scenario-based synthetic data for test and evaluation of information analytics software. *In Proc. Conference on Beyond Time and Errors : Novel Evaluation Methods For information Visualization, 2008, 1-9.*, 2008.

C.D. Wickens, D. L. John, Y. Liu et S.E.G. Becker : *An Introduction to Human Factors Engineering.* Upper Saddle River, NJ : Pearson Prentice Hall, 2 édition, 2004.

L. Williams : *Beginning XSLT and XPath : Trans-forming XML Documents and Data.* John Wiley & Sons Ltd, 2009.

M.A. Winckler, P. Palanque et C. Freitas : Tasks and scenario-based evaluation of information visualization techniques. *In Proceedings of the 3rd annual conference on Task models and diagrams*, TAMODIA '04, pages 165–172, New York, USA, 2004. ACM. ISBN 1-59593-000-0. URL http://doi.acm.org/10.1145/1045446.1045475.

D. Wixon : Evaluating usability methods : why the current literature fails the practitioner. *interactions*, 10(4):28–34, 2003. ISSN 1072-5520. URL http://doi.acm.org/10.1145/838830.838870.

D. Woo et J. Mori : Accessibility : A tool for usability evaluation. *In Computer Human Interaction, 6th Asia Pacific Conference, APCHI 2004, Masoodian, Jones, Rogers. Springer-Verlag, Rotorua, New Zealand*, pages 531–539, 2004.

P. Wright, M. Blythe, J. McCarthy, S. Gilroy et M. Harrison : User experience and the idea of design in hci. interactive systems. design, specification, and verification. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, 1:1–14, 2006.

J. Xiong : *Une méthode d'inspection automatique de recommandations ergonomiques.* Thèse de doctorat, the University of Toulouse, 2008.

J. Xiong, M. Diouf, C. Farenc et M. Winckler : Automating Guidelines Inspection : From Web site Specification to Deployment. *In Computer-Aided Design of User Interfaces (CADUI), Bucharest, Romania, 05/06/06-08/06/06*, pages 0–0, http ://www.springerlink.com/, 2006. Springer-Verlag.

R. Yvonne, H. Sharp et J. Preece : *Interaction Design : Beyond Human-Computer Interaction.* Wiley Publishing, 3 édition, 2011.

L.S. Zettlemoyer, R. Amant et M.S. Dulberg : Ibots : agent control through the user interface. *In Proceedings of the 4th international conference on Intelligent user*

*interfaces*, IUI '99, pages 31–37, New York, USA, 1999. ACM. ISBN 1-58113-098-8. URL http://doi.acm.org/10.1145/291080.291087.

Z. Zhang, V. Basili et S. Shneiderman : Perspective-based usability inspection : An empirical validation of efficacy. *Empirical Software Engineering : An International Journal*, 4:43–70, 1999.

# Cinquième partie

# Annexes

Annexe A

# Règles ergonomiques utilisées pour l'évaluation expérimentale

1. **R1.** La densité d'affichage des données ne doit pas être exagérée. (Tinker, 1965) La densité d'affichage globale ne doit pas franchir les 30% (Danchak, 1976).
2. **R2.** La taille de la police utilisée doit être de 12 points pour les menus et de 10 points pour les fenêtres (Galitz, 2007).
3. **R3.** Utiliser des fontes d'écritures simples et connues afin d'assurer une lecture aussi rapide que possible (Arial, Verdana Sans Serif, Times New Roman ou Georgia Serif) (Galitz, 2007).
4. **R4.** Il est préférable de garder le couleur d'arrière plan du système (Galitz, 2007)
5. **R5.** Toute liste de sélection ou de combinaison, défilante ou non défilante, graphique ou non peut être remplacée par une liste déroulante équivalente si la densité de l'affichage est élevée ou si la préférence de sélection de l'utilisateur est élevée ou si l'utilisateur est expérimenté (Kim et Foley Don, 1990).
6. **R6.** Eviter de surcharger les interfaces avec trop de couleurs. Compte tenu de l'imprévisibilité des écrans couleur et des utilisateurs, le choix peut devenir très compliqué. La couleur est souvent le mieux utilisé pour mettre en évidence les informations clé. En règle générale, ne pas utiliser plus que trois couleurs primaires pour obtenir des informations (Watzman, 2002)
7. **R7.** Les dimensions de tout objet graphique doivent être suffisamment grandes pour être visibles à l'utilisateur et ne doivent pas être petites pour ne pas générer de problèmes d'utilisation (Scott MacKenzie, 1992)

8. **R8.** Optimiser le nombre d'éléments sur une interface pour qu'elle soit Claire. Se restreindre vous moyennement sur une vingtaine de composants graphiques (Galitz, 2007).
9. **R9.** Toutes les fenêtres doivent avoir un titre affiché en haut. Ce titre doit être descriptif de l'interface. Une exception pour les messages d'erreurs (Galitz, 2007).
10. **R10.** Les champs texte doivent être remplis au préalable afin de mieux guider l'utilisateur (Galitz, 2007).
11. **R11.** L'interface doit fournir à l'utilisateur des mécanismes pour revenir dans l'état précédant (Contrôle explicite) (Bastien et Scapin, 1993).

Annexe B

# Questionnaire utilisé pour l'évaluation expérimentale

1. Avez-vous pu réaliser ce que vous avez voulu faire ?
   - Oui
   - Non
   - Je ne sais pas
2. Il y avait-il quelque chose dont vous vous y attendez dans l'interface et vous n'avez pas trouvé ?
   - Oui
   - Non
3. Comment jugez-vous la qualité de l'interface ?
   - Excellente
   - Bonne
   - Moyenne
4. Comment estimez-vous le vocabulaire utilisé par le système ?
   - Compréhensible
   - Acceptable
   - Difficile
5. L'exécution de chaque tâche est t elle bien décrite ?
   - Oui
   - Non
   - Je ne sais pas

6. Les actions sont elles possibles autrement que par la souris ?
   - Oui
   - Non
   - Je ne sais pas
7. Le système propose t-il des notifications lors de l'exécution d'une tâche ?
   - Oui
   - Non
   - Je ne sais pas
8. Le système informe t-il l'utilisateur concernant les tâches qu'ils sont en train d'exécuter ?
   - Oui
   - Non
   - Je ne sais pas
9. La navigation entre les interfaces est elle cohérente et intuitive ?
   - Oui
   - Non
   - Je ne sais pas
10. Les messages d'erreurs sont ils claires et faciles à comprendre ?
    - Oui
    - Non
    - Je ne sais pas
11. Comment trouvez-vous la documentation et/ou l'aide proposées ?
    - Excellente
    - Acceptable
    - Pas claire
12. Estimez-vous qu'il y a un souci de cohérence en termes de notations entre le SAI et les systèmes interactifs usuellement utilisés
    - Oui
    - Non
    - Je ne sais pas
13. Estimez-vous qu'il y a un souci de cohérence entre les différents items proposés dans le système ?
    - Oui
    - Non
    - Je ne sais pas

14. Il y a t-il des mécanismes permettant de recouvrir aux erreurs d'utilisation ?
    - Oui
    - Non
    - Je ne sais pas
15. Le système présente t-il des mécanismes permettant de le modifier suivant vos propres besoins ?
    - Oui
    - Non
    - Je ne sais pas
16. Comment estimez vous les informations présentées par le système ?
    - Oui
    - Non
    - Je ne sais pas
17. Le fonctionnement de l'interface est t-il bien facile à comprendre ?
    - Oui
    - Non
    - Je ne sais pas
18. Le fonctionnement de l'interface est t-il bien facile à apprendre ?
    - Oui
    - Non
    - Je ne sais pas
19. Avez-vous des difficultés à manipuler l'interface ?
    - Oui
    - Non
    - Je ne sais pas
20. Durant l'interaction vous vous êtes senti rapide ?
    - Oui
    - Non
    - Je ne sais pas
21. Avez-vous fait des manipulations inutiles ?
    - Oui
    - Non
    - Je ne sais pas

Annexe C

# Synthèse sur des outils pour l'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques

| Outil | Entrée | | | | Sortie | |
|---|---|---|---|---|---|---|
| | Données | | Données requises | IU évaluée | Service fourni | Evaluation |
| | G | I | | | | |
| Synop | X | | Fichier binaire généré par Imagin | WIMP | RE non respectées, suggestions d'amélioration et correction automatique de l'IU | Statique |
| Ocawa | X | | Parseur de code | WUI | RE non respectées | Statique |
| TAW | X | | Parseur de code | WUI | RE non respectées | Statique |
| Sherlock | X | | Fichier descriptif de l'IU | WIMP | RE non respectées | Statique |
| Ergoval | X | | Description des attributs de l'IU | WIMP | RE non respectées | Statique |
| Destine | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Magenta | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Takata | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Eval Access | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| WAEX | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Chimes | X | | Manuel | WUI | RE non respectées | Statique |
| Kri/Ag | X | | Description textuelle | WIMP | RE non respectées | Statique |
| Gibbons | X | | Questionnaire | WIMP | RE non respectées | Statique |
| Weblint | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Sherlock (Mahajan et Shneiderman, 1997) | X | | Un module Translator qui convertit la source suivant un format canonique | WIMP | Résultats d'analyse telles que la densité des IU et des statistiques | Statique |
| Hyper AT | X | X | Parseur de code HTML et Fichier Log | WUI | Analyse structurelle | Stat et Dyn |
| Bobby | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Doctor Watson | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Reweb / Testweb | X | X | Parseur de code HTML et UML | WUI | RE non respectées | Stat et Dyn |
| A-Prompt | X | | Parseur de code HTML | WUI | RE non respectées (HTML et Earl) | Statique |
| Access Enable | X | | Robot | WUI | RE non respectées et suggestions d'amélioration | Statique |
| Webxact | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| HTML Toolbox | X | | Parseur de code HTML | WUI | RE non respectées et suggestions d'amélioration | Statique |
| Lift Machine | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Websat | X | | Parseur de code HTML | WUI | RE non respectées | Statique |
| Google Analytics | X | | Parseur de code HTML | WUI | Les problèmes d'accessibilité | Statique |
| AwebHut | X | | Parseur de code HTML | WUI | deux rapports d'évaluation (global et détaillé) | Statique |
| ErgoCoIN | X | | Questionnaire | WUI | Problémes de conception de l'IU | Statique |

TABLE C.1: Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connsaisances ergonomiques (Partie 1)

| Outil | Dimension | | | Automatisation | | | Flexibilité | Outil |
|---|---|---|---|---|---|---|---|---|
| | Utilité | Utilisabilité | Accessibilité | Capture | Analyse | Critique | | |
| Synop | X | X | | A | A | A | Choix des RE | Logiciel |
| Ocawa | | | X | A | A | A | Rapport d'évaluation et RE | SW [1] |
| TAW | X | | X | A | A | M | Pas de codage en dur | Logiciel et SW |
| Sherlock | | X | | A | M | M | Choix des RE | Client-serveur |
| Ergoval | | X | X | M | A | A | Rapport d'évaluation et RE | Logiciel |
| Destine | | X | X | A | A | A | Pas de codage en dur des RE | Logiciel |
| Magenta | | | X | A | A | A | Pas de codage en dur des RE | Logiciel |
| Takata | | | X | A | A | M | Sélection de RE | Logiciel |
| Eval Access | | | X | A | A | A | Pas de codage en dur des RE | SW |
| WAEX | | | X | A | A | M | Pas de codage en dur des RE | SW |
| Chimes | | X | | M | A | A | Aucune | Logiciel |
| Kri/Ag | X | | | M | A | M | Sélection de RE | Logiciel |
| Gibbons | X | | X | M | A | M | Sélection de RE | Logiciel et SW |
| Weblint | X | X | X | A | A | M | Sélection de RE | Logiciel et SW |
| Sherlock (Mahajan et Shneiderman, 1997) | | X | X | A | A | M | Aucune | Logiciel |
| Hyper AT | | X | | A | A | A | Sélection de RE | Logiciel |
| Bobby | X | | X | A | A | M | Selection de RE | Logiciel |
| Doctor Watson | | X | | A | A | M | Selection de Tests | SW |
| Reweb et Testweb | | X | | S | A | M | Aucune | Logiciel |
| A-Prompt | | | X | A | A | M | Selection de Tests | SW |
| Access Enable | | | X | A | A | A | Pas de codage en dur | SW |
| Webxact | | | X | A | A | A | Aucune | SW |
| HTML Toolbox | | X | | A | A | A | Aucune | Logiciel |
| Lift Machine | | X | X | A | A | A | Sélection de RE et rapport | Logiciel |
| Websat | | | X | A | A | M | Sélection de RE | Logiciel |
| Google Analytics | | | X | A | A | A | Aucune | SW |
| AwebHut | X | | | A | A | A | Aucune | SW |
| ErgoCoIN | X | | | A | A | A | Aucune | SW |

TABLE C.2: Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques (Partie 2)

| Outil | Intervenants | Connaissances requises de l'évaluateur | Gestion des RE |
|---|---|---|---|
| Synop | Evaluateur et expert en IHM | Ergonomie logicielle et en Imagin | Système expert |
| Ocawa | Evaluateur | Aucune | W3C/WAI, WCAG et l'ADAE/DGME |
| TAW | Evaluateur | HTML et ergonomie logicielle | Mobile OK beta, WCAG 1.0 et WCAG 2.0 |
| Sherlock | Evaluateur, concepteur et un expert en ergonomie et en utilisabilité | Avancé | Le serveur stocke les RE ainsi que les algorithmes de validation |
| Ergoval | Evaluateur | Basique | Aucune |
| Destine | Evaluateur | HTML | Aucune |
| Magenta | Evaluateur | Ergonomie Logicielle | Outil intégré : The Guideline Editor |
| Takata | Evaluateur | Aucune | WCAG 1.0 (SGSL) |
| Eval Access | Evaluateur | Aucune | WCAG 1.0 |
| WAEX | Evaluateur | Aucune | WAI, W3C et Mobile Ok Basic |
| Chimes | Evaluateur | Aucune | OSF/MOTIF |
| Kri/Ag | Evaluateur | Aucune | 70 RE |
| Gibbons | Evaluateur et utilisateurs | Aucune | 400 RE |
| Weblint | Evaluateur et utilisateurs | Aucune | Aucune |
| Sherlock (Mahajan et Shneiderman, 1997) | Evaluateur | Les RE ont été sélectionnée par un groupe de chercheurs à l'Université de Maryland (USA) | Aucune |
| Hyper AT | Evaluateur | Aucune | Aucune |
| Bobby | Evaluateur | Aucune | WAI et 508 Section |
| Doctor Watson | Evaluateur | Aucune | Aucune |
| Reweb et Testweb | Evaluateur | Aucune | Aucune |
| A-Prompt | Evaluateur | Ergonomie du Web | WCAG 1.0 et WCAG 2.0 |
| Access Enable | Evaluateur | Aucune | 508 Section et WCAG 1.0 |
| Webxact | Evaluateur | Aucune | WCAG 1.0 |
| HTML Toolbox | | Aucune | Analyses syntaxiques, linguistiques et structurales |
| Lift Machine | Evaluateur | Aucune | WCAG 1.0 et 508 Section (140 RE) |
| Websat | Evaluateur | Aucune | ISO 9242 |
| Google Analytics | Evaluateur | Aucune | Test d'accessibilité |
| AwebHut | Evaluateur | Aucune | |
| ErgoCoIN | Concepteur et utilisateurs | RE sélectionnées par l'auteur | |

TABLE C.3: Tableau récapitulatif des outils d'évaluation des interfaces utilisateur exploitant les connaissances ergonomiques (Partie 3)

Annexe D

# Synthèse sur des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'interface et l'utilisateur

| **Outil** | **Entrée** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Application** | | | **Type** | | **Intervenants** | | |
| | **WIMP** | **WUI** | **Mobile** | **WebService** | **Logiciel** | **Concepteur** | **Utilisateur** | **Evaluateur** |
| EMA | X | | | | X | | X | X |
| USINE | X | | | | X | X | X | X |
| RemUsine | | X | | X | | | X | |
| Wet | | X | | X | | | X | X |
| iBOT | X | | | | X | | X | X |
| Web Quilt | | X | | X | | | X | X |
| SMT | | X | | X | | | X | X |
| Web RemUsine | | X | | | X | | X | X |
| MESIA | X | | | | X | X | X | X |
| MM Web RemUsine | | X | | | X | X | X | X |
| Click Tale | | X | | X | | X | X | X |
| Multi Device RemUsine | | | X | | X | | X | X |
| Capture and Reply Tool | X | | | | X | X | X | X |
| Click Heat | | X | | X | | | X | X |
| Morae | | X | | | X | | X | X |
| EISEval | X | | | | X | | X | X |
| Feng Gui | | X | | | X | X | X | X |
| Crazy Egg | | X | | X | | | X | X |
| Clixpy | | X | | X | | | X | X |
| Hallway | | X | | X | | | X | X |
| Usabilla | | X | | X | | | X | X |
| Userfly | | X | | X | | | X | X |
| EVALYZER | | X | | X | | | X | X |
| Loop11 | | X | | X | | | X | X |
| Silverback | | X | | | X | | X | X |

TABLE D.1: Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 1)

| **Outil** | **Evaluation** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Principe d'évaluation** | **Evenements Capturées** | | | | | | **Evaluation** | |
| | | **But** | **Domaine** | **Niveau Abstrait** | **Evt IU** | **Evt entrée** | **Physique** | **Précoce** | **Test** |
| EMA | comparaison | X | X | X | X | X | X | | X |
| USINE | comparaison | | | X | X | X | X | | X |
| RemUsine | comparaison | | | X | X | X | X | | X |
| Wet | Données numériques | | | | | | X | | X |
| iBOT | Observation | | | | | | X | | X |
| Web Quilt | Observation | | | X | | | | | X |
| SMT | Mouvement de la souris | | | | | | X | | X |
| Web RemUsine | Comparaison | | | X | X | X | X | | X |
| MESIA | Comparaison | | | | | | X | | X |
| MM Web RemUsine | Observation | | | X | X | X | X | | X |
| Click Tale | Mouvement de la souris | | | | | | X | | X |
| Multi Device RemUsine | Observation interaction | | | X | X | X | X | | X |
| Capture and Reply Tool | Observation | | | | | | X | | X |
| Click Heat | Mouvement de la souris | | | | | | X | | X |
| Morae | Observation | | | | | | X | | X |
| EISEval | Comparaison | | | X | X | | X | | X |
| Feng Gui | Eye Tracking | | | | | | | | X |
| Crazy Egg | Mouvement de la souris | | | | | | X | | X |
| Clixpy | Observation | | | | | | X | | X |
| Hallway | Observation | | | | | | X | | X |
| Usabilla | Observation | | | | | | X | | X |
| Userfly | Observation | | | | | | X | | X |
| EVALYZER | Collecte des données relatives au comportement de l'utilisateur d'une façon exhaustive | X | X | X | X | X | X | | X |
| Loop11 | Analyse de statistiques | | | X | X | X | X | | X |
| Silverback | Observation | | | | | | X | | X |

TABLE D.2: Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 2)

| **Outil** | | | | |
|---|---|---|---|---|
| | **Technique de capture** | **Automatisation** | | |
| | | **Capture** | **Analyse** | **Critique** |
| EMA | Injection de code dans la source | A | A | M |
| USINE | Log file (QC/Replay and JDK) | A | A | M |
| RemUsine | Log file (QC/Replay and JDK) | A | A | M |
| Wet | Event Handling | A | A | M |
| iBOT | Capture d'écran | A | A | M |
| Web Quilt | Proxy | A | A | M |
| SMT | Script | A | A | M |
| Web RemUsine | Fichier Log | A | A | M |
| MESIA | Injection de code dans la source | A | A | M |
| MM Web RemUsine | Fichier Log, eye tracking et Webcam | A | S | M |
| Click Tale | Script | A | A | M |
| Multi Device RemUsine | Fichier Log | A | A | M |
| Capture and Reply Tool | Fichier log | A | A | M |
| Click Heat | Injection de code Java script | A | A | M |
| Morae | Webcam et microphone | A | A | M |
| EISEval | Injection de code dans la source | A | A | M |
| Feng Gui | Capture d'écran | A | M | M |
| Crazy Egg | Injection de code Java script dans la source | A | A | M |
| Clixpy | Injection de script dans la source | A | A | M |
| Hallway | NA | A | A | M |
| Usabilla | Injection de code dans la source | A | A | M |
| Userfly | Injection de code Java script | A | A | M |
| EVALYZER | Acquisition vidéo | A | A | M |
| Loop11 | Questionnaire | A | A | M |
| Silverback | NA | A | A | M |

TABLE D.3: Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 3)

| **Outil** | **Sortie** | | |
|---|---|---|---|
| | **Interactivité** | **Fonctionalités** | **Rapport** |
| EMA | NA | détecter le(s) rupture(s), le(s) annulation(s) immédiate(s), la(s) répétition(s) et l'invalidité dans la séquence de manipulation de l'interface | NA |
| USINE | NA | données numériques sur le taux d'accomplissement des tâches | Aucun |
| RemUsine | NA | données numériques sur le taux d'accomplissement des tâches | Aucun |
| Wet | NA | information portant essentiellement sur le chargement des pages ainsi sur la navigation entre les pages | Aucun |
| iBOT | Aucun | Identifier les actions répétitives, inutiles ou erronées | Aucun |
| Web Quilt | Aucun | renseigne sur un ensemble de données concernant la navigation entre les pages web (temps de consultation, fréquence, etc.). | Oui |
| SMT | Oui | Permettre de tracer le mouvement de la souris sur l'interface | |
| Web RemUsine | NA | Indiquer des données numériques sur le taux d'accomplissement des tâches | Aucun |
| MESIA | Non | Identification des problèmes et des difficultés d'utilisation de l'interface | Aucun |
| MM Web RemUsine | NA | renseigne sur les actions de l'utilisateur interagissant avec l'IU avec un ensemble de données (mouvement des yeux, actions élémentaires, etc. | NA Aucun |
| Click Tale | Oui | localiser les clikcs, leur nombre, le mouvement de la souris | 2 Rapport |
| Multi Device RemUsine | NA | renseigne sur les actions de l'utilisateur interagissant avec une interface Mobile | Aucun |
| Capture and Reply Tool | NA | Identifier les actions intitules et les tâches non exécutées. | NA |
| Click Heat | Oui | nombre de click ainsi que leur localisation | Oui |
| Morae | Oui | Permet d'indiquer des informations portant sur les séquences d'interaction entre l'IU et l'utilisateur | Oui |
| EISEval | Non | Identification des problèmes et des difficultés d'utilisation de l'interface | Aucun |
| Feng Gui | Oui | Une heat map de l'IU | Oui |
| Crazy Egg | Oui | Mouvement de la souris | NA |
| Clixpy | Oui | Une base de données des actions utilisateurs lors de la session de Test | Oui |
| Hallway | Oui | La séquences d'actions lors d'une session d'interaction entre l'IU et l'utilisateur | Non |
| Usabilla | Oui | Une heat map des clics | Oui |
| Userfly | Oui | Enregistrement vidéo de la session d'interaction | Non |
| EVALYZER | Oui | Capture tâche utilisateur | Oui |
| Loop11 | Oui | Rapport des problèmes détectés | XML, PDF, etc. |
| Silverback | Oui | Vidéo de la session d'interaction | Vidéo |

TABLE D.4: Tableau récapitulatif des outils d'évaluation des IHM basés sur la capture, l'étude et l'analyse de l'interaction entre l'IU et l'utilisateur (Partie 4)

**Résumé**

Ce mémoire contribue à la thématique de l'évaluation des interfaces utilisateur. Il se base sur deux contributions pour l'évaluation de ces interfaces. La première consiste en un Framework se basant sur trois différentes techniques d'évaluation (le mouchard électronique, le questionnaire et l'inspection des règles ergonomiques). La seconde consiste en deux ensembles de composants graphiques pour la composition d'interfaces graphiques avec les environnements de développement intégré du type WYSIWYG. Le premier ensemble vise à inspecter la présentation statique de l'interface lors de la conception de cette dernière. Cette inspection s'effectue localement par le composant graphique lui-même par rapport à un ensemble de règles ergonomiques. Le second ensemble consiste à l'évaluation de l'interface utilisateur en analysant l'interaction entre l'utilisateur et l'interface. Cet ensemble de composants opère avec un mouchard dédié à cet usage. Une évaluation de ces deux contributions ainsi que des perspectives de recherche concluront ce présent mémoire.

**Mots-Clés.** Interfaces Homme-Machine, Interface Utilisateur, Interaction Homme-Machine, évaluation, techniques d'évaluation, mouchard électronique, règle ergonomique.

---

**Abstract**

This thesis report is a contribution to the user interface evaluation theme. Our thesis is based mainly on two contributions for user interface evaluation. The first one consists of a Framework dedicated for user interface evaluation. It is based on three evaluation techniques (the electronic informer, the questionnaire and the ergonomic inspection). The second consists of two sets of graphical controls for composing the user interface using WYSIWYG integrated development environment. The first set is intended to inspect the user interface static display during the design phase. This inspection is elaborated by the control itself according to ergonomic guidelines. The second set consists in evaluating user interfaces by analyzing the interaction between the user and the interface. This set of controls operates with an electronic informer dedicated to user interface evaluation. An evaluation of these two contributions as well as research perspectives concludes this present report.

**Keywords.** Human-Computer Interaction, User interface, evaluation, evaluation techniques, electronic informer, ergonomic guidelines.