Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

Discipline ou spécialité :

Informatique et Télécommunications

Présentée et soutenue par :
Jorge Ricardo GÓMEZ MONTALVO

le : jeudi 16 février 2012

Titre :

A Multimedia Ontology-Driven Architecture for Autonomic Quality of Service Management in Home Networks

---

JURY

M. Christophe CHASSOT, Président
M. Elie NAJM, Examinateur

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
Laboratoire d'Analyse et d'Architecture de Systèmes

Directeur(s) de Thèse :

M. Ernesto EXPOSITO, M. Michel DIAZ

Rapporteurs :
M. Olivier DUGEON
M. Flavio OQUENDO

A Jorge Adrián,
La fuerza detrás de todo, mi orgullo más grande.

A mi papá†.

A mi mamá.
Mi maestra, mi amiga, mi modelo a seguir, el cimiento de todo lo que tengo en la vida.

Canek habló a Guy:
- Mira el cielo; cuenta las estrellas.
- No se pueden contar.
Canek volvió a decir:
- Mira la tierra; cuenta los granos de arena.
- No se pueden contar.
Canek dijo entonces:
- Aunque no se conozca, existe el número de las estrellas y el número de los granos de arena. Pero lo que existe y no se puede contar y se siente aquí dentro, exige una palabra para decirlo. Esta palabra, en este caso, sería inmensidad. Es como una palabra húmeda de misterio. Con ella no se necesita contar ni las estrellas ni los granos de arena. Hemos cambiado el conocimiento por la emoción: que es también una manera de penetrar en la verdad de las cosas.

– *Ermilo Abreu Gómez*

Acknowledgments

Indeed, what I have printed out in this dissertation is what rest forever after several years of research work. But here, in the Acknowledgments, I printed out what rest on me forever. Thanks to all of you.

I have made my thesis as a member of the team named *Services et Architectures pour les Réseaux Avancés* (SARA) which former name is *Outils et Logiciels pour la Communication* (OLC) of the *Laboratoire d'Analyse et d'Architecture des Systèmes* (LAAS-CNRS). I would like to thank to LAAS, its director M. Arlat and former director M. Chatila. Also, I would like to thank to M. Vernadat and M. Drira, OLC and SARA directors respectively, for accepting me to work in this honorable community. I really appreciate all the support they gave me.

I would like to thank to *Facultad de Matemáticas de la Universidad Autónoma de Yucatán* and to *PROgrama de MEjoramiento del Profesorado* (PROMEP) for giving me the financial support during all these years. I have the moral commitment of working hard for my University. I have many friends from Mérida Yucatán that at the time, they encouraged me and helped me to be here: Fernando Curi, Luis Basto, Roger Castillo, Francisco Moo, Julio Díaz, Lucy Torres, Luis Rodríguez y Francisco Madera. Gracias por el apoyo brindado.

I would like to express my deepest gratitude to my thesis advisor Ernesto Exposito, whose ideas allow me to develop this thesis. He was not only my mentor but a friend, a colleague, who helped me in many ways. I guess I would have made two or three Ph.D. thesis during this time if I would have been able to develop my work as fast as he produces new interesting ideas. *Muchas gracias Ernesto, tuve al mejor asesor*.

I would also like to thank my thesis co-advisor Michel Diaz. His experience allow me to drive my efforts in the best direction at several times during these years. I have learned from him the way french people work, I have took the best from that and it helped me a lot when I applied it to my research work. I have really appreciated his observations on my manuscript as well as I have enjoyed the travels for the Feel@Home meetings. *Merci beaucoup* Michel.

I would like to thank to the other members of jury: *M. le président du jury* Christophe Chassot, Olivier Dugeon, Elie Najm, Flavio Oquendo. Thanks for taking the time for reading and evaluating my thesis. I have appreciated your comments about my dissertation.

I would like to thank the Feel@Home partners. For me, it was more than another great experience in my professional development. Thanks for giving me this opportunity. Olivier Dugeon, Mahdi Mohammed, Marta, Romain Carbou, David, Manuel, Gema, Remi, Julio, and to all of the Feel@Home colleagues, thank you very much.

When someone is far from his home, with a little bit of good luck, one is able to make friends and groups of friends who are like small families. Fortunately, that was my case, and I have been adopted by several families. Now, thanks to all of you, I have different ways of seeing life. There is no doubt that all of you contribute to make my life richer.

I would like to thank to the family integrated by the Mexican Boys: Guillermo, Manuel, José Luis, Raúl, Eduardo. All the vacations and pique-niques that we share were wonderful. Mario et Dora, Francisco Montecillos, Gilberto, Iván et Valerie, Stéphane Ávila y esposa, Bernardo y Ana, fue un placer conocerles.

My friends of LAAS, (almost) in order of appearance: Florin Racaru, M. Alberdi, Nicolas, Roxana, Batispte, Germán Sancho and Nadia, Ismail et Emna, Riad Ben Halima, Rodrigo Saad, Akram, Ihsane Tou, Ixbalank, Sandy, Lionel, Denis, M. Kremer, M. Dugue, Pedro Casas, Sakku, Codé, M. Juanole *et sa révolution*. Denis, thanks a lot for helping me with my french *résumé, ouuuuf! ce n'est pas facile le francais!* Codé, thanks again for helping me during my preparation for my Ph.D. defense. I would also like to thank to Papa and Moctar, without their help I would not have obtained good outcomes, there is a part of their internship work in this dissertation.

I would like to thank to my tunisian friends Aymen Kamoun, Emna Mezghani, Hatem Arous, Mohammed Amine, Ghada Gharbi et Mahdi Ben Alaya, Nouha Abid and Rim Akout, Mohammed Siala. A3chik barcha. Naraft tawa ana Jorge el tounsi. Allahu Akbar.

My french family Jeff Tissot, Sandrine, Casey, they allow me to see the warmest side of french people. Special thanks to Jeff and his band of *copains*: Bruno, Sandrine et Arnaud. I've appreciated the VTT lessons that you gave me. To ride a bicycle is like to be a child again.

Edgar y Myriam, no pude escoger mejores hermanos. Mis hermanos Dante y Sofía. Ustedes son mi amada familia. No tienen idea de lo importante que son para mi. Tenerles a mi lado me ha hecho un hermano muy presumido y muy orgulloso de ustedes. Gracias por el apoyo en momentos difíciles.

Jorge Adrián, tú has sido el motor, la fuerza que me condujo a terminar este trabajo. Sin ti, seguramente no habría terminado. Gracias.

Last but no least, I would like to thank to my Mijnouna Miziena, tu m'as apporté beaucoup de choses dans ma vie, mais surtout tu m'as donné une nouvelle manière de vivre la vie. Tu m'as appris la patience, tu m'as donné la paix et la calme. Tu m'as montré la joie dans tous les moments dont nous avons profités ensemble. Tu

as su me motiver quand je ne pouvais plus me motiver. Tu m'as mis dans le chemin qui m'emmène là, où je le veux. Je veux bien parcourir ce chemin avec toi tout ma vie. Je veux bien te remercier pour tout mais je sais que ça ne suffit pas. Nhibik barcha.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 Motivation

IN the last years there have been considerable changes in network communications. To mention some of them, one can find an increasing number of new network users demanding the ability of being *on-line* from a wide variety of devices like Personal Computers, Laptops, Personal Digital Assistants, mobile phones, etc. Moreover, there have been changes in user requirements such as staying "plugged" to a network anywhere, anytime using any communication device. These user requirements have boosted the use of wireless networks, but also the complexity of network management and the implementation of new protocols or services considering new network characteristics such as traffic variability.

Nowadays, we are closer to a world where "specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence" as stated by Mark Weiser when he presented his vision of ubiquitous computing in [Wei91].

In this pervasive, ubiquitous, all-interconnected world, Distributed Multimedia Systems (DMS) have become a kind of communication system that is a "must-have" for the users. Every day, it is more and more common to use services like Voice over IP (VoIP), video-calls from mobile phones (connected to WiFi or 3G), IPTV, Video On Demand (VoD), or classic video conference systems. Besides, the use of social networks has clearly increased the exchange of multimedia information between users.

There are different scenarios where one can find these new user requirements all together. For example, a person using its mobile phone for watching a video from *youtube* while he is waiting at the airport. At home, a kid playing video games against other on-line players and against his visiting friend while his sister is watching a movie on demand at the TV, and at the same time, his father is having a talk using a video conference. Certainly, in this kind of context, an adequate management of shared network resources helps to provide certain quality for DMS.

QoS management for time-constrained multimedia applications has been always a challenge, mostly when they run over Best-Effort networks [BBC$^+$98, BCS94, RVC01]. End-users demand the same QoS for their applications regardless the kind of network they are connected to (WiFi or 3G) [SABW99, SMH07, HKK04]. Depending on their previous experiences, end-users usually expect better results from a service, for example, a better resolution in a video conference or less degradation in the quality of audio and video streaming. In other words, users want to have a better Quality of Experience (QoE)[Uni01a, Uni08a]. Indeed, user participation in QoS management has recently gained attention in the domain of QoE research.

The increasing number of ubiquitous scenarios in which users are able to exchange multimedia information combined with the complexity of communication system management and end-user participation in QoS provisioning have exposed new challenges and created new research areas which claim for new solutions and new communication paradigms.

Our research is focused on the QoS and QoE management for multimedia applications in the context of home networks taking into account the viewpoint of involved actors (end users, network providers, and application programmers). In the next section we present in more detail the context of our work as well the problematic we have identified.

1.2 Context and Problem Statement

As previously mentioned, this research work is situated in the context of home networks, in which it is very interesting the enforcement of the QoS in home scenarios. A natural question that arise is: *why are we motivated for QoS management in home networks?* From our home-network context, we think that, on the one hand, QoS provisioning in Internet is quite complex because:

- It is very difficult to establish the necessary agreements between many different network providers. Certainly, there are several aspects in which network providers must agree before deploying a QoS solution over the Internet, for example, QoS oriented protocols or charging schemes.

- There are so many users, then it will be very difficult to differentiate them when they are sharing network resources.

- There are many applications and new ones everyday, that makes very difficult to treat them according to their requirements.

On the other hand, regarding QoE consideration and QoS provisioning in the context of home networks, we have found that:

- There is usually just one Internet service provider (ISP), thus it is easier to agree about the use of network resources.

- There is a small group of home users, it allows one of them to define a hierarchy or priorities between them as well as to express their QoS/QoE preferences or requirements.

- There are fewer time-constrained applications, thus it is easier to integrate them in QoS provisioning frameworks.

Certainly, in home network scenarios, there are several *types of users*, for example, children, father, mother; or different *user roles* such as administrator, home user, or visiting user. Each *type of user* or *user role* could have a different *priority*; and even also, they could assign different *priorities* to their applications.

As we can see in Figure 1.1, there are also other actors beside home users, such as service providers or Internet service providers whom participation in QoS management is mandatory.

Now, let us imagine that in the scenario presented in Figure 1.1, every user in the home network is running different applications such as large file transfers and multimedia applications. Let us suppose that at a certain moment, the home administrator begins to suffer some degradation in the quality of his video conference.

A natural solution would be that the home administrator cancels every other (multimedia or not) sessions in the network in order to obtain the necessary network resources for his very important video conference. The home administrator could manually cancel or request all other users to cancel their applications. Even if this solution may be effective, we think that would be better if the home administrator would have a means to predefine the network behavior needed in order to maintain the QoS of the priority applications belonging to priority users.

We can see that in such a scenario, the home administrator needs to express his/her preferences, for example, about the importance of users and/or applications. But also, we can notice that it will be necessary that the communication system gets

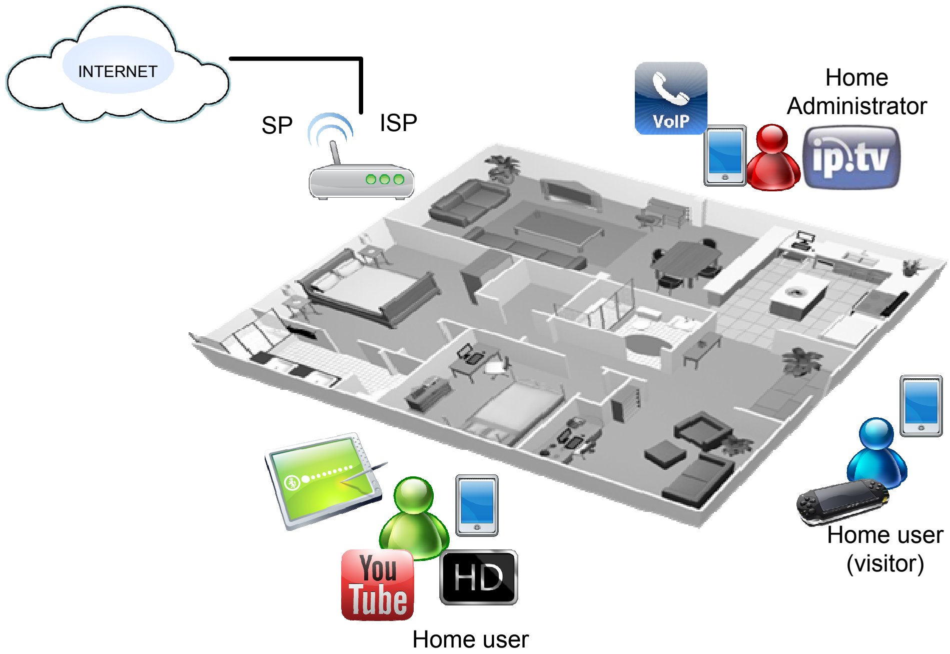

Figure 1.1: Home Network Context

the same understanding of these user preferences. An even more complete solution would allow the application programmers to attach the QoS requirements of multimedia applications in order to ensure a good performance of critical applications.

1.2.1 Semantics

A definition of semantics given by the merriam-webster[1] dictionary is: "the study of meanings" or "signification of words". Certainly, semantics implies the use of a language, words, statements, signs, etc. In the context of home networks, if we want to take into account user preferences in QoS management, it is necessary that the different involved actors have the same understanding about what the users prefer, what the applications need, and what the networks can provide. In other words, semantics plays an important role in our context.

Trying to find a solution, we ask ourselves the following questions:

- How can a user describe his/her preferences about its applications ?
- How can an administrator user describe his/her preferences about other users ?

[1]http://www.merriam-webster.com/

- How can application programmers describe the QoS requirements of multimedia applications ?
- How can the available home network resources be characterized ?
- How to share the semantics of DMS QoS requirements, user QoE preferences, and network resources characterization ?

We have realized that currently there is no available solution to answer these questions. When we try to answer these questions, we can see that it is necessary to describe or characterize, in a computer readable format, the multimedia applications as well as its QoS requirements. Also, it is necessary to model the user preferences and make this information available to a QoS manager entity at the moment when network resources need to be assigned to applications. In other words, we need a framework which is able to represent the knowledge of each actor and to share the same semantics.

We believe that if the users, applications, programmers, and home-network service providers have the mechanisms to describe and communicate their requirements or the services they offer, then we would be able to take advantage of this information and design a generic framework intended to manage the QoS considering all actors viewpoints. Such QoS framework would allow a smarter network resources management.

Semantic frameworks may allow home users to express QoE requirements or preferences. Likewise, semantics should allow designers and programmers to translate and to express user requirements and preferences in terms of QoS requirements. Also, semantics can help the systems to detect when such user QoE requirements or preferences are not compatible with the services offered by service providers. In the same manner, semantics can help designers and developers to design and to develop applications and multimedia services respecting QoS and QoE constraints. Having a way to represent QoS preferences and requirements in computer readable expressions, will allow the system to take them into account in order to efficiently manage network resources or QoS.

In order to express semantics and to represent the knowledge of a domain, ontologies are a suitable approach [GG95, CJB99]. Indeed, ontologies allow a community of users to share the same understanding of the knowledge that they are representing.

After giving the state of art in QoS and QoE in Chapter 2, Chapter 3 presents concepts, definitions, and the state of art in the domain of knowledge representation as well as how ontologies can be used in order to accomplish the objectives of this thesis.

1.2.2 Self-management of QoS

Let us go back to our home network scenario. We are aiming to manage the QoS in a home network but considering user preferences like user or application priorities. Indeed, in spite of the development of QoS oriented protocols at network and transport layers [BBC$^+$98, BCS94, RVC01, IAC99], such provisioning of QoS for DMS is not a trivial task. In a home network, the administrator could manually release the necessary network resources in order to assign them to more important users or critical multimedia applications. Clearly, such approach does not offer an efficient nor pragmatic solution.

Even in the context of home networks, user preferences can dynamically change as well as the priority of applications when other applications are executed. Furthermore, a user can change his/her application priority according to the experience he/she had using it. Certainly, we want to propose a solution that is able to manage the QoS with or without a minimum user intervention. In other words, we are looking for a framework that is able to dynamically adjust itself in order to manage the QoS in a home network. More precisely, we are looking for an autonomic QoS framework that is able to adapt network resources assignation according to user preferences and applications priorities.

Let us suppose that we have already solved knowledge sharing and semantics problems. At this point, we ask ourselves the not simple question: *How to automatically manage network resources based on user preferences and application QoS requirements in a home network?*

As a matter of fact, this question leads us to ask ourselves more questions such as:

- What information is needed in order to self-configure the communication network ?

- How to self-configure the communication network in order to enforce the QoS/QoE requirements and preferences ?

The Autonomic Computing paradigm [Hor01] is an approach that proposes self-properties for autonomous systems. There are four general self properties, namely: Self-configuration, Self-healing, Self-optimization, Self-protection [IBM03]. In an autonomic framework, it is necessary to propose new communications and network paradigms which allow self-* characteristics in communication's architectures, and reduce the network management complexity by allowing seamless integration of user's goals and system constraints. Currently, it is an open problem to manage QoS for DMS in an autonomic approach. In this thesis, we are also focusing on the self-management for providing QoS in home networks.

We have identified, as part of the problems addressed in this thesis, how to develop an autonomic framework which is able to self-manage the QoS for DMS in home networks. We believe that through the use of an ontology-based framework as a source of knowledge representation and semantic sharing (of multimedia systems, QoS requirements, user preferences, etc.), it will be possible to design autonomous network components that implement the required self-management. Chapter 4 presents in detail the concepts, definitions, and the state of art in the autonomic computing domain as well as how this approach will be used to reach the objectives presented in the following section.

1.3 Objectives

To solve the problems introduced in the previous section, let us state the main goal of this research work.

We aim to provide QoS for distributed multimedia systems by developing an ontology-based framework that allows end-users and applications' programmers to express their QoS requirements and preferences; and to characterize communication services (network, transport, middleware) in order to obtain the knowledge needed for providing autonomic QoS Management in the context of Home Networks.

In order to accomplish the above mentioned goal, we propose the following sub-objectives:

- Develop an ontology-based framework that integrates the different viewpoints of the actors (e.g. communication services, users, applications' programmers). With such ontology-based framework, the actors will be able to describe their requirements and communicate them as necessary. Moreover, available services will be characterized, in consequence, they will be used for management purposes.

- Develop an autonomic QoS management architecture, which will allow to enforce accurate decisions in an autonomous manner based on the information provided by the ontology-based framework.

- Evaluate the functionality of the ontology-based framework and the performance of the autonomic QoS management framework.

The challenge we are proposing to solve is the following:

An ontology-based framework can allow end users, application programmers, and home-network providers to describe and communicate their preferences and requirements as well as to characterize available network resources. Based on this *common semantics, an autonomic QoS management driven by user preferences and application requirements in the context of Home Networks can be implemented.*

1.4 Contributions

As we mentioned, in the context of home networks, end user participation in QoS management is an interesting challenge. We have decided to build the solution to the problematic of this thesis in two parts. First, we think that in order to consider user preferences and QoS requirements of multimedia applications, it is necessary to provide a semantic space shared by all the actors. Second, we believe that the proposition of an autonomic approach will allow to manage QoS in home networks while considering user preferences and application requirements.

As consequence, in this thesis, we have developed the following two main contributions:

- We are proposing an ontology-based approach named MODA, which stands for *Multimedia Ontology Driven Architecture*. The main goal of MODA is to allow home users, application programmers, and home-network providers to have a common space, where they can share the semantics about multimedia services, QoS parameters and requirements of multimedia applications.

- We are proposing an autonomic QoS management architecture, which is able to take into account user preferences and application requirements in order to assign home network resources. Based on the knowledge provided by MODA, our approach allows to dynamically integrate all the needed semantic information in order to self-manage home-network resources.

The above mentioned contributions have been implemented and evaluated in the context of the Feel@Home project [BMMMMD11]. Chapter 5 presents the project itself as well as the implementation and evaluation of our contributions.

1.5 Structure of the dissertation

The structure of this dissertation is organized as follows. **Chapter 2** presents a general state of the art in the domains of QoE and QoS. We start by presenting the definitions of QoE and its importance. Also, we present the existent subjective and objective evaluation mechanisms for the QoE. In the second part of this chapter, we present the overview and definition of QoS. The most important QoS frameworks at different levels of the network stack are briefly presented as well as the main concepts and frameworks of QoS in the context of home networks.

Chapter 3 has three main sections. The first section presents a state of the art in Model Driving Architecture and Engineering, more precisely, about the current utilization and support of models in the process of software design, development, and deployment. The second section presents the state of art in the knowledge representation domain. It particularly focuses on ontologies as a means to represent the knowledge of a domain. Also, it describes the Web Ontology Language (OWL) and its properties as a Description Logic Language. The third section presents one of the main contribution of this thesis, an ontology-based framework that we have named Multimedia Ontology Driven Architecture (MODA). MODA aims to provide a space of knowledge shared between all actors in the context of home networks.

Chapter 4 is organized in two main sections. The first section presents a state of the art of Autonomic Computing and Communications. The benefits and advantages of the autonomic computing regarding the challenges of our research are presented. Here, we found the explanation of why the autonomic paradigm suits well our research. The second section presents the second main contribution of this thesis: an autonomic QoS provisioning framework in the context of home networks. In order to show the feasibility of our autonomic approach, a simulation of the autonomic QoS manager is presented as a proof of concept.

Chapter 5 describes, in two main sections, the deployment and evaluation of our contributions: autonomic QoS management architecture based on MODA. The first section presents the Feel@Home project, which provides our study context and contains the QoS scenarios in which we are interested. The second section presents the deployment and evaluation of our work. A case study of a Video-on-Demand application is presented in order to show how to instantiate the ontologies included in MODA, as well as how to create a complete session description of such multimedia applications including its non-functional aspects. Then the section presents how to actually enforce the autonomic QoS provisioning in a home network scenario. At the end of this chapter, the results obtained from our tests are presented and analyzed.

Finally, the general conclusion of this thesis is presented as well as several research perspectives.

CHAPTER 2
State of the Art

Contents

2.1 Introduction

In the last years, there have been considerable changes in the manner in which users utilize the network as a means of communication. The wide deployment of Internet has facilitated the apparition of scenarios where home-users can communicate using multimedia applications such as VoIP or videoconferencing. Furthermore, the current use of social networks has increased the apparition of dynamic multimedia communication scenarios in which users can easily create instantaneous multimedia sessions.

New user requirements such as staying plugged to a network anywhere, anytime and using different devices have boosted the use of wireless networks, including home Wi-Fi networks. Moreover, these scenarios have facilitated the exchange of multimedia information between mobile users. Indeed, one consequence of the apparition of these new user requirements is that the complexity of network resources management has augmented.

Also, the complexity involved in heterogeneous network scenarios (i.e., broadband Internet, wired and Wi-Fi networks) and the diversity of user requirements (i.e., being connected and communicating all the time, from different networked devices, etc.) have originated the apparition of pervasive and all-interconnected ubiquitous Distributed Multimedia Systems (DMS). In everyday life, it is more and more common use new services like VoIP, mobile video-calls, television over IP, Video on Demand, video conferences systems, etc.

In a pragmatic world, QoS provisioning for distributed multimedia applications is mandatory. We can assume that in (almost) any scenario, the user wants to have the best possible QoS for his/her multimedia sessions, for example, video conferences free of freezing images, VoIP with a quality at least as good as in traditional phone calls, real-time responses when he/she is playing interactive online video games, and so on. In the domain of QoS management, QoS is provided following two general approaches: by over-provisioning of network resources or by QoS control mechanisms, for example, admission and traffic control.

Over-provisioning, as its name suggests, provides enough network resources to avoid the lack of QoS. For example, capacity over-provisioning tries to avoid potential overloads by using capacity dimensioning methods to define the required network capacity [MMC06]. Some other works such as [SD10] propose provisioning or dimensioning of network resources based on analytical models in order to achieve the desired QoS. However, these works usually address Internet backbones and enterprise IP networks. Thus, they usually do not consider some factors that are present in local area networks such as resources-limited home networks and that can have an impact on the QoS observed by the final user.

On the other hand, there are works such as [Nah99] which support the idea that no matter the over-provision of network resources, it is still necessary to manage network resources in order to guarantee certain QoS. There are several QoS models at the different levels of the network stack with the goal of providing QoS. For example, at network layer there are DiffServ [BBC$^+$98] and IntServ [BCS94]; at MAC layer, there are standards like standards IEEE 802.1Q [IEE06] and IEEE 802.11e [IEE05]. Section 2.3 explains in detail such QoS solutions.

Nowadays, in the context of home networks, the predominant approach to provide QoS is over-provisioning. Indeed, there are a few works that propose and develop QoS management in home networks. The UPnP Forum[1] has proposed the UPnP QoS Architecture [For08c] which has become the *de facto* reference for

[1] http://www.upnp.org/

QoS management in home networks. UPnP QoS Architecture includes user priorities and the specifications for its enforcement. However, this specification is usually not implemented in real home networks scenarios.

Regardless if over-provisioning of home-network resources is used or not, we envisage situations in which the available network resources are not sufficient to satisfy all the applications requirements. Then, user preferences should be taken into account to manage the QoS. Moreover, we think that end-users should have the means that allow them to manage the network resources based on the quality they have experienced about an application. Certainly, by taking into account the opinion the user has about the QoS offered by an application, it is possible to provide better QoS as well as to close the feedback loop between the service provider and the consumer of the service.

This chapter introduces QoS concepts, particularly in the context of Home Networks (HN) and Distributed Multimedia Systems in order to give to the reader the fundamental basis to understand the problems related to QoS provisioning. Also, the chapter presents several multimedia-related protocols used at different levels of the IP stack aimed at illustrating how QoS requirements are expressed by distributed multimedia applications. In the final part of this chapter, we introduce current QoS management solutions in the context of Home Networks.

2.2 Quality of Experience

The user perception of the provided quality is a subject that recently has gained a lot of attention in the QoS domain. By nature, the user perception is subjective, as it can change as the user context change, or simply because two different users may have a different Quality of Experience from the same service because each user has different expectations of that service. The following subsection presents the main concepts in QoE, most of them introduced by the International Telecommunication Union (ITU)[2].

2.2.1 What is QoE and why it is important

ITU-T Recommendation G.1000 [Uni01a] proposes four different viewpoints of QoS. Such viewpoints of QoS include two actors: the user (customer of a service) and the service provider. The four viewpoints are defined as follows:

- QoS requirements of user. A statement of the level of quality required by the applications of users of a service, which may be expressed non-technically.

[2] http://www.itu.int/ITU-T/

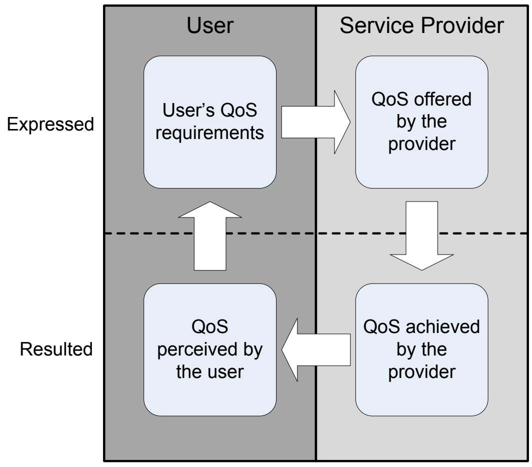

Figure 2.1: The different viewpoints of QoS (ITU-T G.1000).

- QoS offered by provider. A statement of the level of quality expected to be offered to the user by the service provider.
- QoS delivered by provider. A statement of the level of the actual quality achieved and delivered to the user.
- QoS perceived by user. A statement expressing the level of quality that users believe they have experienced.

As we can see in Figure 2.1, the ITU-T Rec. G.1000 [Uni01a] proposes the concept of *QoS perceived by the user* and defines it, in other words, as: "the statement expressing the level of quality experienced *he believes* he has experienced".

Later, the ITU Focus Group on IPTV [Uni08a] has defined Quality of Experience (QoE) as: "the overall acceptability of an application or service, as perceived by the end-user".

The latter definition has attached two important considerations: first, QoE includes the complete end-to-end system effects; second, overall acceptability may be influenced by user expectations and contexts. One can realize that QoE is inherently subjective, and thus it is not an easy task to assess the quality that a user has experienced from a service.

2.2.2 Evaluation Mechanisms of the QoE

ITU has proposed recommendations of subjective and objective evaluation mechanisms of the QoE. In spite of ITU's efforts, there are no widely deployed mechanisms to evaluate the QoE. This can be explained because, on the one hand, the application of subjective evaluation mechanisms is time consuming, as they should be applied to a considerable number of users, or simply because it is not easy to reproduce the necessary conditions in which such mechanisms should be conducted.

On the other hand, objective evaluation mechanisms are based on complex mathematical models, usually not easy to understand, thus making difficult their application and result interpretation by people who do not have such mathematical skills. However, objective evaluation mechanisms have the advantage that they put aside subjective inputs (introduced by users opinions).

The following subsection presents the main subjective evaluation mechanisms of the QoE.

2.2.2.1 Subjective Mechanisms

In order to subjectively assess the user perception of quality, different metrics and methods have been proposed as a manner to map user opinion to numerical-objectives values. As a matter of fact, ITU-T Rec. G.1000 [Uni01a] mentions, as an example, that a user may assign a number value on a “5-point” scale to its service experience, where “5” indicates an excellent service and “1” means *bad quality*. Ideally, there would be “1:1” correspondence between delivered and perceived QoS.

ITU-T Recommendation P.800 [Uni96] describes methods and procedures for conducting subjective evaluations of transmission quality for telephone communication (audio communication). It proposes the *Mean Opinion Score* (MOS) , one of the most popular metrics that assigns numerical values to perceived quality for audio communication. MOS is based on an *Absolute Category Rating* (ACR) , that is a test method in which subjects are asked to express opinion judgements using absolute quality scales, for example, excellent, good, etc. [Uni06a]

Recommendation ITU-T P.800.1 [Uni06b] presents MOS in the following way: “mean of opinion scores, is the mean of the values on a predefined scale that subjects assign to their opinion of the performance of the telephone transmission system used either for conversation or for listening to spoken material”.

We can say that the principle of MOS is simple, it was drawn as an example in the ITU-T Rec. G.1000: we ask users to assign a “*grade*”, ranging from 1(bad) to 5(good), to a service. After that, the mean of such grades (MOS) is calculated in order to obtain a “*general*” opinion of the QoS experienced by the user. Table 2.1 shows the “5-point” listen quality scale used to obtain the MOS.

Even if MOS was defined for its utilization in audio communications, the same idea was adopted to measure the user perception of quality in different kinds of multimedia applications such as video-based applications. Recommendation ITU-T P.910 [Uni08d] describes non-interactive subjective assessment methods for evaluating the one-way overall video quality for multimedia applications such as video-

MOS	Quality of Speech
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 2.1: Listen Quality Scale

conferencing, storage and retrieval applications, telemedical applications, etc.

The main problem with MOS and with other subjective evaluation mechanisms of the QoE, is that there are external or environmental factors to the evaluated service that may influence the user's subjective opinion. In order to isolate the measurements of MOS from environmental factors, ITU has proposed a set of methods to obtain subjective evaluations of transmission systems and components. One disadvantage of these methods is that they are conducted under scenarios with strict setup conditions. Naturally, these conditions are difficult to reproduce in a common context like a home network. Furthermore, in order to obtain reliable MOS results, it is necessary to apply the tests to a considerable number of users of the communication system under evaluation.

2.2.2.2 Objective Mechanisms

The goal of using objective evaluation mechanisms is to replace subjective user opinions by *predicting* what the users would say from a communication service. Recommendation ITU-T P.800.1 [Uni06b] presents other MOS related scores, some of them, obtained from objective models and algorithms.

In order to calculate MOS in an objective manner, mathematical models and algorithms consider only the inherent parameters and factors of the communication system, i.e., acoustical environment, environmental noise, transmission errors, packet loss, transcoding, etc. The success of an objective evaluation mechanism depends on how well its prediction is correlated to subjective results. Table 2.2 shows objective, subjective and predicted MOS nomenclature for listening only, conversational, and talking systems.

Other objective mechanisms have been defined in order to predict the quality perceived by a user in terms of MOS. In the case of perceptual video quality measurements, Perceptual Evaluation of Video Quality (PEVQ)[Uni08c] has gained a lot of attention. PEVQ is a robust and complex model which is designed to predict the effects of transmission impairments on the video quality as perceived by a human subject. Its main targets are mobile applications and multimedia applications.

	Listening only	**Conversational**	**Talking**
Subjective	MOS-Listening Quality Subjective	MOS-Conversational Quality Subjective	MOS-Talking Quality Subjective
Objective	MOS-Listening Quality Objective	MOS-Conversational Quality Objective	MOS-Talking Quality Objective
Estimated	MOS-Listening Quality Estimated	MOS-Conversational Quality Estimated	MOS-Talking Quality Estimated

Table 2.2: MOS Terminology

PEVQ is built on Perceptual Video Quality Measure (PVQM) [HBL$^+$02], a TV quality measure developed by KPN Research[3] and Swisscom Innovations[4].

2.2.3 Summary

This section has presented the QoE concept and its evaluation mechanisms. At first, this section presents several QoE definitions provided by different international standards as an effort to have a wide accepted QoE understanding. From these definitions we have realized that, in fact, there are two spaces from where quality of a service is observed: the customer and service provider viewpoints. We think that it is necessary to semantically align these two viewpoints.

This section has also presented the mechanisms used to evaluate the QoE. There are subjective and objective evaluation mechanisms of the QoE. Mean Opinion Score is one of the most popular subjective evaluation mechanisms for QoE. The weakest point of subjective mechanisms resides in the inherent subjectivity of customer opinions. Customer's opinion may depend and vary according to many external factors e.g., previous service experiences, environmental factors, humor, etc. In order to reduce subjectivity, objective mechanisms for QoE evaluation have been proposed by several international standards. They try to predict what customers would say from a communication service. Such prediction is based on the mathematical analysis of inherent parameters of the communication system.

Unfortunately, QoE is usually not considered as an input to drive the QoS management. The vast majority of QoS provision solutions traditionally consider only network mechanisms or parameters to maintain QoS requirements of applications. The following subsection presents the main solutions in the domain of QoS provisioning.

[3]http://www.kpn.com/
[4]http://www.swisscom.ch

2.3 Quality of Service (QoS)

Several studies have been carried out to define and characterize the notion of Quality of Service. QoS has been defined by the ITU-T Recommendation X.902 [Uni09] as the "set of qualities related to the collective behavior of one or more objects". More specifically in the area of information technology and multimedia systems, the Quality of Service has been defined as "the set of quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application" [VKvBG95].

2.3.1 What is QoS and why it is important ?

In spite of the diversity of approaches, most of them agree on defining the QoS from the user and service provider viewpoints. The ITU-T Recommendation E.800 [Uni94] explicitly introduces the user/service approach and defines the QoS as "the collective effect of service performance, which determine the degree of satisfaction of a user of the service".

On the one hand, from the user's viewpoint, user requirements express the quantitative and qualitative characteristics expected from a particular service; these requirements can generally be expressed in terms of QoS parameters (e.g., time constraints, synchronization, throughput, reliability, order, etc.). On the other hand, from the service provider's viewpoint, the QoS is considered as a statement of the level of quality expected to be offered to the user of the service.

In [Uni01a], the QoS concept is enhanced by integrating two temporal phases: an initial phase where requirements and services can be expressed and a following phase where the delivered QoS can be observed. (See Figure 2.1 in section 2.2.1). There are 4 points of view illustrating the semantic complexity involved in QoS provisioning. Any QoS solution has to provide a consistent semantic translation between these different views, during both the top-down initial phase and the bottom-up operational phase (i.e. measurements and performance evaluation).

Moreover, the notion of QoS can evolve when new requirements appear; the context change; or new services are deployed. For this reason, static provisioning techniques cannot ensure the necessary adaptation of QoS-oriented systems.

A framework where users could easily express their requirements and preferences and where they also could provide feedback about the service obtained is required for an autonomous QoS provisioning system. In such framework, the quality experienced by the users (QoE) and its translation to the QoS that can be provided by the communication system (i.e. QoS at transport, network and link layers) need to be integrated in a standard and common semantic model. In that way, such a framework is difficult to design, but it should be easily adopted to manage QoS in well identified domains such as home networks (i.e. UPnP QoS architecture).

2.3.2 QoS in distributed systems

In order to provide a complete End-to-End QoS, several QoS solutions have been proposed at the different layers of the network stack. The following subsections presents the main QoS solutions at application, transport, network and link layers.

2.3.2.1 QoS at application layer

In the context of distributed multimedia applications, QoS mechanisms at the application layer mainly focus on the encoding and/or decoding process. Indeed, depending on the used media codec, the audio-video streams may require different transmission rates to provide a good quality at the destination.

Regarding the compression quality, codecs may be classified as lossy or lossless codecs. On the one hand, lossy codecs compress the original multimedia information to be stored or transmitted but, as a result of compression, they lose information and some of the original quality of the media. On the other hand, lossless codecs keep the original quality of the media in spite of the compression they provide.

Assigning the necessary bandwidth to a codec's transmission rate is critical, mostly for real-time, near-realtime or unbuffered multimedia applications. Thus, QoS requirements cannot be satisfied when the available bandwidth is not enough for a codec transmission rate. Certainly, the codec must be designed to support a constant bit-rate (CBR) or variable bit-rate (VBR) transmission, but also to respect the nature of the media being transmitted. For example, CBR fits better for voice transmissions; and VBR suits better for video transmissions.

Adaptive encoding is one of the most used mechanism for QoS control with codecs. Research work like [JKK10, KKK+11] propose mechanisms for QoS control based on such adaptive encoding techniques, for example, by controlling the codec bit-rate according to network state e.g., network congestion, available bandwidth, packet loss ratio, etc. However, in order to use this kind of QoS mechanisms, a codec capable of encoding the information at different bit-rates is needed. Other work like [NHS05] propose to renegotiate the initial SIP session establishment [HSSR99] using the SDP protocol [HJP06] in order to adapt the codec parameters according to the information provided by RTCP protocol [SCFJ03].

2.3.2.2 QoS at Transport Layer

In both OSI and TCP/IP reference models, the transport layer provides end-to-end communication services. In the OSI reference model, the transport layer aims at providing transparent transfer of data between the communicating systems and relieve the users of the transport service from the details of using the available network services [Uni95]. Transport protocols implement the required functions operating over the network services in order to offer the required transport services to the OSI session layer. On the other hand, in the original design of the TCP/IP model, the two primary transport protocols that were considered are TCP and UDP.

QoS at the transport level can be expressed by a least the following set of requirements:

- Reliability. Packet loss rate (PLR) tolerance.
- Order. Out of sequence tolerance.
- Throughput. Transmission capacity per time unit.
- Delay. End to end transmission time.
- Jitter. Variation of the delay.

In order to provide a better QoS, the above mentioned parameters can be affected, among others, by the following mechanisms [IAC99]:

- Error Control. Set of mechanisms that deals with loss or damage of user data and control information. Error control is done in two phases: error detection, and error reporting and recovery. Error detection deals with lost, disordered, duplicated, and corrupted Transport Protocol Data Units. During error reporting, the transport receiver explicitly informs the transport sender about detected error. Error recovering includes data retransmission or redundancy data mechanisms in order to recover from errors.
- Flow and Congestion Control. Flow Control is a mechanism used by the sender to limit the rate at which data is sent over the network in order to avoid buffer overflow in the receiver. Congestion control or congestion avoidance mechanisms prevent from sending too much traffic in the underlaying network.
- Multiplexing and Demultiplexing. Multiplexing allows multiple transport layer connections to be associated to a single network layer e.g., several transport layer ports to a single network address in a connectionless oriented network. Thus, multiplexing uses network layer resources more efficiently by reducing the network layer's context-state information. Demultiplexing mechanisms are necessary at the transport receiver if multiplexing is used.
- Segmentation and Reassembly. Segmentation occurs when it is necessary for the transport sender to divide a Transport Service Data Unit (TSDU) into

smaller Transport Protocol Data Units (TPDU) because the network service imposes a maximum permitted Network Service Data Unit (NSDU). The transport receiver collects the TPDU's to rebuild the original TSDU to be delivered to the user receiver.

2.3.2.3 QoS at Network Layer

The initial service offered by the IP networks was the Best-Effort service, characterized by no guarantee in the delivery of data packets. Best-Effort is still the predominant service in the Internet. During the past decades, two approaches have been proposed for QoS provisioning in Internet. The first approach asserts that over-provisioning the network resources as result of their low prices will allow QoS to be automatically delivered. The second approach affirms that no matter how much bandwidth the networks can provide, new applications will be invented to consume it. Furthermore, the current state of the Internet characterized by limitations in network resources leads to support the second approach and to continue the research in the development of QoS provision mechanisms.

The IETF has proposed several services models intended to control and manage the QoS at the network layer. In this section the Integrated Services (IntServ) [BCS94] and the differentiated services (DiffServ) [BBC+98] models will be mentioned. Next paragraphs present these models, the services offered, the mechanisms supported and the advantages and disadvantages of each one.

Integrated Services (IntServ). In this model, QoS is guaranteed by the preliminary reservation of network resources for every data flow [BCS94]. The transmission of a data flow is preceded by the configuration and reservation of resources in the nodes placed in the path between sending and receiving applications. The Resource Reservation Protocol (RSVP) is a signaling protocol used for applications to reserve resources in IntServ domains [BZB+97].

The main disadvantage of the IntServ model is the important amount of flow-related state information that has to be maintained in every router. The requirement of storage and processing capabilities is particularly important in the Internet backbone where hundreds of thousands of flows may be present. For this reason, this architecture does not scale well in the Internet. Several extensions have been proposed to use RSVP for the reservation of flow aggregates in order to reduce the scalability problem. The integration of IntServ with other models (i.e. DiffServ model), aimed at providing an end-to-end QoS architecture has been proposed in [BFY+00].

Differentiated Services (DiffServ). The DiffServ model has been proposed by the IETF in order to solve the scalability problems of IntServ to provide QoS at the network layer [BBC+98]. DiffServ is based on two fundamental design principles: the complexity is pushed to the network boundary (i.e. applications, leaf routers or edge routers) and the policing and supporting mechanisms are implemented separately [ZW00].

First principle is based on the fact that network boundaries process a relative small number of flows and therefore they can perform more efficiently complex operations such as packet classification and traffic conditioning. In contrast, a network core router may have an important number of flows to process and for this reason it should perform fast and simple operations.

The separation of operations between the boundary and the core permits the scalability of the DiffServ model. The second principle allows control policy and supporting mechanisms to evolve independently. In DiffServ, several per-hop packet forwarding behaviors are defined as the basic building blocks for QoS provisioning. The control policy can be changed as needed, thus assuring the flexibility of the DiffServ model. The DiffServ model proposes two classes of services:

- Expedited forwarding (EF): The EF service provides low delay and low jitter service for customers that generate fixed peak bit rate traffic. This service is also called the premium service. Traffic exceeding the service contract will be discarded. A service contract is also called a Service Level Agreement (SLA).

- Assured forwarding (AF): this service provides four different service classes and, for every class, three drop precedence levels can be defined. Every DiffServ router reserves minimum resources for every AF class (i.e. buffer space, bandwidth, etc), but the classes can receive more resources than demanded if available. When network is congested, the drop precedence level can be used to perform selective discarding.

Other QoS-oriented network services. In the current Internet, the complexity involved in the deployment of the QoS network models is such that most of current Internet users are only able to access traditional Best-Effort networks, where QoS requirements are not guaranteed. In the case of IntServ, the drawback of this model is that it is necessary to maintain flow status information in each node along the path, with leads to an enormous management overhead. The scalability problems of IntServ have led the proposition of the DiffServ model where the resources are not managed per flow but aggregating flows into classes, thereby reducing the management overhead. Other mechanisms based on Multi Protocol Label Switching (MPLS) [RVC01] or Traffic Engineering (i.e. constraint-based routing) [AMA+99] have also been proposed to improve packets forwarding through the Internet. Other proposals based on peer-to-peer and overlay networks are also under development.

2.3.2.4 QoS at Link Layer

There are several efforts that aim to provide QoS at link layer. The IEEE computer society has proposed standards in this domain. In the following paragraphs, two main IEEE standards for QoS provisioning at link layer are briefly explained.

IEEE Standard 802.1Q. The IEEE standard 802.1Q [IEE06] defines the operation of virtual bridged local area networks or VLANs. Basically, this standard adds a tag header to the frame containing all the necessary information in order to differentiate the type of frame. IEEE 802.1Q defines two types of tagged frames: VLAN-tagged frames and Priority-tagged frames. The tag header of a priority-tagged frame contains priority information, but carries no VLAN identification information. A VLAN-tagged frame contains both VLAN identification information and priority information. Thus, the specification allows adding priority information (priority of traffic) to frames. This is the reason why we are interested on the priority information of the IEEE 802.1Q tag header.

The IEEE 802.1Q tag header may vary according to the MAC method used to transmit the frame. However, in any case, the tag header includes a:

- Tag Protocol Identifier: it is used to identify an IEEE 802.1Q tagged frame.
- Tag Control Information, which contains a Priority Code Point field (PCP) of 3 bits, a Canonical Format indicator of one bit, and VLAN Identifier of 12 bits (see Figure 2.2).

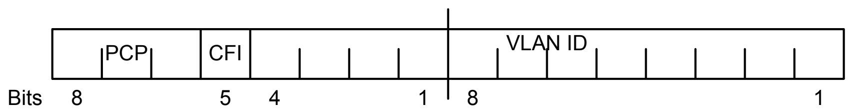

Figure 2.2: Tag Control Information Format (IEEE 802.1Q)

The 3 bits of the PCP field helps to assign priorities to eight different types of traffic. The annex G of the IEEE 803.1Q, revision of 2005, proposes the following traffic classification by default:

- Network control (NC): it is the kind of traffic with guaranteed delivery requirement to support network configuration and maintenance.
- Internetwork control (IC): this is the traffic whose requirement is to distinguish the traffic supporting the network as a concatenation of several administrative domains from the network control traffic of the immediate domain.
- Voice traffic (VO). As stated in the ITU-T Recommendation G.1010 [Uni01b], this kind of traffic has requirements like 10ms delay and, hence, maximum jitter.

- Video traffic(VI): it is the type of traffic for applications with 100ms delay as primary requirement.

- Critical applications (CA): this is the kind of traffic that has a minimum bandwidth as primary requirement but is working along with control admission mechanisms in order to avoid resource consumption abuse at the expense of other applications.

- Excellent effort (EE): the type of traffic generated by the most important users in the network.

- Best effort (BE): the classic best effort traffic generated by unprioritized applications regulated by TCP congestion and error mechanisms.

- Background (BK): bulk traffic permitted on the network but that should have no impact on others applications.

The semantics utilized for the mapping of traffic types to traffic classes actually depends on the number of traffic class queues supported by the compliant IEEE 802.1Q device. Table 2.3 shows the semantic mapping proposed by the IEEE standard 802.1Q: each pair of braces represents a traffic class queue supported by the device, where a pair of braces (queue) contains the traffic types assigned to that queue. The different traffic types assigned to the same queue are treated as equals i.e., with the same priority.

In order to establish the priority values for the traffic types, IEEE 802.1Q proposes the correspondence illustrated in Table 2.4. By default all the traffic transmitted by end stations has a priority value of 0; and the default traffic type is best effort, hence 0 is used for both default priority and for best effort.

IEEE Standard 802.11e. In order to cope with the lack of QoS services in wireless networks, the IEEE 802.11e standard proposes QoS features and multimedia support. The IEEE Standard 802.11e defines the Medium Access Control (MAC) procedure to support LAN applications with QoS requirements like audio, voice, video transport over IEEE 802.11 wireless LAN.

Basically, IEEE standard 802.11e defines two mechanisms to support applications with QoS requirements. The first mechanism is the Enhanced Distributed Channel Access (EDCA) mechanism, which provides a prioritized QoS. By using EDCA, it is possible to deliver traffic based on differentiating user priorities. The second mechanism is the Hybrid Coordination Function (HCF) Controlled Channel Access (HCCA) which allows the reservation of Transmission Opportunities (TXOP) for station hosting applications with QoS requirements. A TXOP is an interval of time when a QoS station can transmit as many frames as possible onto the medium. In this sense, a TXOP has a started time and a maximum duration.

Number of Queues	Traffic Types per queue
1	{Best Effort, Background, Excellent effort, Critical Applications, Voice, Video, Internetwork Control, Network Control}
2	{Best Effort, Background, Excellent effort, Critical Applications} {Voice, Video, Internetwork Control, Network Control}
3	{Best Effort, Background, Excellent effort, Critical Applications} {Voice, Video} {Network Control, Internetwork Control}
4	{Best Effort, Background} {Critical Applications, Excellent effort} {Voice, Video} {Network Control, Internetwork Control}
5	{Best Effort, Background} {Critical Applications, Excellent effort} {Voice, Video} {Internetwork Control} {Network Control}
6	{Background} {Best Effort} {Critical Applications, Excellent effort} {Voice, Video} {Internetwork Control} {Network Control}
7	{Background} {Best Effort} {Excellent effort} {Critical Applications} {Voice, Video} {Internetwork Control} {Network Control}
8	{Background} {Best Effort} {Excellent effort} {Critical Applications} {Video} {Voice} {Internetwork Control} {Network Control}

Table 2.3: IEEE 802.1Q traffic type to traffic class mapping

In an IEEE 802.11e WLAN, the traffic with high priority has more opportunity of being transmitted than traffic with lower priority. Moreover, the duration of TXOP for Video and Voice traffic classes is higher than Background and Best Effort traffic classes.

IEEE 802.11e uses Access Categories (AC) in order to semantically express the different traffic classes. Four traffic classes are considered in the AC, which are illustrated in Table 2.5 as well as the mapping between IEEE 802.1D user priorities correspondence and the AC.

2.3.3 Summary

This section has presented the main efforts to provide QoS at the different layers of the communication stack. Each communication layer has a vision of the QoS, but all of them are focused on user requirements accomplishment. However, QoS mechanisms proposed by one layer usually do not know the QoS mechanisms of other layers. In other words, QoS mechanisms usually work in a isolated manner, without taking into account QoS information of upper or lower communication layers.

We realized that some of the QoS solutions propose the use of priorities in order to differentiate traffic classes. Some of them use the term “user priority” as a refer-

Priority	Acronym	Traffic Type
1	BK	Background
0(default)	BE	Best Effort
2	EE	Excellent Effort
3	CA	Critical Application
4	VI	Video
5	VO	Voice
6	IC	Internetwork control
7	NC	Network control

Table 2.4: Traffic types, Acronyms, and priority values

Priority	User Priority as in IEEE 802.1D	802.1D Description	AC	Description
Lowest	1	BK	AC_BK	Background
	2	-	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	BE	AC_BE	Best Effort
	4	CL	AC_VI	Video
	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
Higher	7	NC	AC_VO	Voice

Table 2.5: 802.1D user priorities to AC mappings

ence to the priority of the traffic class. However, this "user priority" actually means priority of a traffic class which is predefined according to the type of traffic generated by the application (audio, video, network control information). Moreover, the user does not participate in the definition of such traffic class priorities.

We think that in local area networks, the user participation in the definition of priorities of traffic classes is feasible. As a matter of fact, the UPnP Forum has proposed the UPnP QoS Architecture [For08c] which specified the utilization of user priorities for user themselves and also traffic priorities for application flows. The work done by the UPnP Forum has served as basis for other organizations like the Digital Living Network Alliance (DLNA) [All07]. Both solutions are explained in detail in the following section.

2.4 QoS in Home Networks

As explained in previous sections, QoS management in Internet is a complex task. Indeed, there are a lot of service providers that must agree on the QoS oriented protocols in order to provide QoS to end users. In local area networks like home networks, these requirements are not present or they are less complex: for example, there is a small group of service providers and usually there is just one network provider. Furthermore, in a home network there is a reduced number of users, usually the people (family) that lives in there. In consequence, we believe that, in home networks, participation of end users in QoS management is a feasible task but it is not easy to implement.

We think that in current and future home network scenarios, user participation in QoS management is a requirement for any QoS solution for home networks. Indeed, end user pays for a service, thus it is important to provide QoS solutions that satisfy multimedia application requirements of home users.

As it was mentioned before, with the apparition of broadband Internet in home networks, QoS was provided by overprovisioning of network resources particularly bandwidth. Efforts of organizations like the UPnP QoS Architecture [For08c], Home Gateway Initiative (HGI) [Ini06], and Digital Living Network Alliance [All07] have proposed solutions, which offer QoS mechanisms in home networks. Recently, ITU has proposed the Recommendation ITU-T H.622 [Uni08b], which describes the architectural framework of a home network that supports multimedia services.

ITU-T H.622 states that there are two methods to provide QoS in IP networks (IP home networks):

- Class-based QoS. In class-based QoS, traffic is aggregated into a small number of classes of traffic, typically 4-8 classes. The packets marked with the same traffic class obtain the same treatment at the network devices regardless of the application data that they contain. It is less complex because it does not require a session establishment (nor signaling protocol) for the corresponding QoS treatment of the packets, as the packets are tagged with their corresponding priority and they are treated accordingly. Thus, even the packets that are tagged can be retransmitted by network devices that are unable to understand the priority information. Another advantage of class-based QoS is that network devices functioning under the class-based QoS mechanism are not required to maintain the session status of each application such as call set-up and teardown. As an example, HGI proposes a type of class-based QoS.

- In session-based QoS, a session is established if every terminal or application can reserve the necessary resources by a signalling mechanism. When the resource reservation process has succeeded, the transmission quality of the associated service is guaranteed. However, session-based QoS has the following problems: some network devices are unaware of signalling protocol, network devices need a complicated mechanism, and additional session set-up time is introduced by the resource reservation process. As an example, UPnP QoS proposes a type of session-based QoS.

The following subsection presents in more detail the UPnP QoS Architecture.

2.4.1 UPnP QoS Architecture

In the particular context of multimedia, the UPnP Audio Visual (AV) specification [For08a] has defined a set of UPnP devices and service templates that specifically targets devices interacting with entertainment contents such as movies, pieces of music, still images, etc.

Three main logical entities constitute the AV architecture: media server, media render, and control point. The media servers have access to multimedia contents and can send them to other UPnP device via the network. Media renderers are able to receive external content from the network and present it on its local hardware. Finally, the control points coordinate the overall operation and provide the interface to the end-user. Of course, a number of intermediate networked devices such as layer-2 bridges or application level proxies can be used between the two end-points.

Due to the possible competition between various multimedia flows within home networks as well as the limited nature of network resources, a best effort delivery cannot guarantee satisfactory user experience. Particularly in the context of AV, flows are real-time constrained and bandwidth consuming. Distributed games are even more sensitive in terms of jitter and delay.

The UPnP QoS Architecture specification [For08c] has been proposed to manage the QoS for traffic streams flowing between a source and a sink device within a home network. This specification also supports QoS management on the LAN for traffic streams originating from or terminating in a WAN. A traffic stream is viewed as a uni-directional flow from a source device to a sink device, possibly passing through intermediate devices. The UPnP QoS Architecture defines three services: QoS device, QoS policy holder and QoS manager. Interactions between these services and other actors such as users and control points are depicted in Figure 2.3.

The following are the phases involved in UPnP QoS provisioning process:

0. An authorized user (e.g., the user administrating the network) provides the policies to be applied within the UPnP network (e.g., priorities for users and applications over the network resources). In the UPnP specification, no further details are provided about the way users can provide information about priorities and preferences. This is one of the features not easy to abstract and difficult to be implemented.

1. The user launches an application requesting QoS from the network. The control point is requested by the application and information about the user and

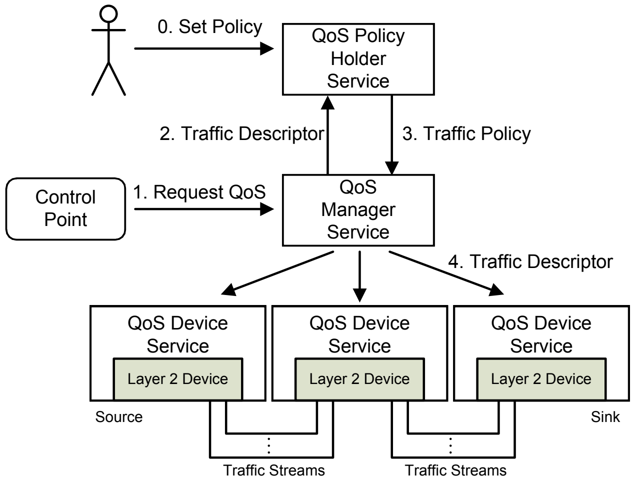

Figure 2.3: The UPnP QoS architecture (V3.0)

the traffic (e.g., source, sink, content, etc.) is provided to the control point. The control point makes a QoS request to the QoS manager service. The control point constructs a traffic descriptor structure and requests to the QoS manager service of the UPnP network to setup QoS for the traffic stream.

2. The QoS manager requests the traffic policy from the QoS policy holder by providing the traffic descriptor.

3. The QoS policy holder uses stored policies in order to provide the traffic policy to the QoS manager. The traffic policy structure includes the traffic importance number and a user importance number.

4. The QoS Manager configures the QoS device services for establishing the QoS for the new traffic stream. The QoS devices derive the specific layer 2 priority to be applied to the traffic stream according to their technical specification. If not enough resources are available for the new traffic stream, the QoS policy holder service can be requested to provide the user importance number for every blocking traffic stream (i.e. previous admitted traffic streams), and if necessary, a preemption process can be performed by taking away resources from blocking traffic streams in order to admit the new traffic stream. The UPnP standard does not consider taking into account users' feedback in the QoS management process. However, user satisfaction feedback could largely improve the QoS management process.

2.4.2 Digital Living Network Alliance (DLNA)

DLNA[5] is an organization of enterprises of leading consumer electronics, computing industry and mobile device companies looking for making possible to end users to enjoy its digital information from any place in their homes. In order to achieve such a goal, DLNA proposes guidelines and certifications to interconnect a wide variety of home electronic devices such as PC, printers, network storage units, set-top boxes, mobile phones, TV displays, game consols, etc. Indeed, DLNA aims at allowing home users to transparently share not only the home network but also the digital content that exists in this environment.

Transparent interoperability of DLNA devices is a key feature. Regarding discovery and control of devices, as well as media management, DLNA is based on the UPnP Device Architecture, UPnP AV, and UPnP Print Enhanced specifications. In DLNA networks, the media is transported using HTTP. As well as in UPnP networks, DLNA supports IPv4 addressing of devices interconnected over WiFi, Ethernet, and Coaxial Cable.

DLNA proposes three device categories [All07], namely: Home Network Devices (HND), Mobile Handheld Devices (MHD), and Home Interoperability Devices (HID). In turns, HND's includes five classes of devices, namely:

- Digital Media Server (DMS): devices providing media acquisition, recording, storage, and sourcing capabilities. DMS products will often include Digital Media Player (DMP) capabilities and may have intelligence, such as device and user services management, rich user interfaces and media management, aggregation and distribution functions.

- Digital Media Player (DMP): devices finding content exposed by a DMS or Mobile-DMS and then pulling the selected content from the server to provide playback and rendering capabilities.

- Digital Media Renderer (DMR): devices playing content received from a DMS or M-DMS after being setup by another Home Network Device (DMC or devices which include a DMC).

- Digital Media Controller (DMC): devices finding content exposed by a DMS and matching it to the rendering capabilities of a DMR, setting up the connections between the DMS and DMR.

- Digital Media Printer (DMPr): devices providing printing services to the DLNA home network. Photo printing is the application DLNA priority, but other types of content can also be printed from a DMPr.

[5]http://www.dlna.org

MHD category includes five device classes. They are similar to HND devices but because of their mobility, they have different media format and network connectivity. They include the following devices:

- Mobile Digital Media Server (M-DMS): it exposes and distributes content.
- Mobile Digital Media Player (M-DMP): it finds content exposed by an M-DMS or DMS and plays the content locally on the M-DMP.
- Mobile Digital Media Uploader (M-DMU): it sends content to an M-DMS or DMS with upload functionality.
- Mobile Digital Media Downloader (M-DMD): it finds and downloads content exposed by an M-DMS or DMS and plays the content locally on the M-DMD after download.
- Mobile Digital Media Controller (M-DMC): it finds content exposed by an M-DMS or DMS and matches it to the rendering capabilities of a DMR, setting up the connections between the server and renderer.

Finally, the HID includes two device classes:

- Mobile Network Connectivity Function (M-NCF): it provides a bridging function between the MHD network connectivity and the HND network connectivity.
- Media Interoperability Unit (MIU): it provides content transformation between required media formats for the HND Device Category and the MHD Device Category.

In DLNA networks, QoS provisioning are based on the UPnP QoS Architecture, i.e., DLNA provides a static mapping between the Differentiated Services Code Point (DSCP) field at IP layer and a associated priority of the traffic class in the data link layer.

2.4.3 Summary

This section has presented the main QoS solutions for Home Networks. The UPnP QoS Architecture of the UPnP Forum has provided the basis for other organizations that propose QoS solutions for home networks. These efforts show that QoS is important and QoS management in home networks is not the exception. However, current QoS solutions for home networks propose mainly a static mapping of application traffic to traffic classes.

The UPnP QoS Architecture has innovated with its proposition of real user priorities. Indeed, the UPnP QoS Manager is able to take into account user priorities.

However, the semantics of such priorities and its mapping to traffic treatment are not specified. Also, in home networks, priorities are currently considered only for traffic flows as specified in layer 2 standards and not for actual user priorities.

2.5 Chapter Conclusion

This chapter has presented the QoS concepts that are the basis of our research work. The chapter has presented in three main sections the following domains: QoE, QoS and QoS in home networks. The first section presents the definition of QoE, how it is measured, why it is important, as well as its relationship with Quality of Service provisioning.

The chapter also has presented the QoS definitions from the point of view of two actors: the end user (customer) and the service provider. The different perspectives from which QoS can be defined help us to realize the several meanings that QoS may have regarding the QoS the user wants and the QoS he receives, and also, regarding the QoS the service provider offers and the QoS it actually delivers.

The state of art presented in this Chapter allows us to understand the complexity of QoS provisioning at application layer, transport layer, network layer, and link layer. Also, one can see that there is no semantic framework in the existent QoS solutions allowing the integration between end user and service provider perspectives. The lack of semantic frameworks includes the particular context of Home Networks where QoS mechanisms are necessary to provide a correct performance of multimedia applications. Thus, the mandatory questions are: 1) how to develop a semantic space shared by home users, application programmers, and service providers that is able to align QoS preferences and requirements between them? and 2) how to integrate such semantic common space in a QoS solution that is driven by user preferences?

The following two chapters answer these two questions. Chapter 3 presents the design and development of a semantic common space which enables sharing knowledge between the involved actors. The semantic framework should be able to characterize in an unambiguous and computer readable manner, all the concepts and its relationships that are necessary for a QoS provisioning driven by semantic models. Chapter 4 presents an answer to the second question. It presents the integration of the proposed semantic framework with an autonomous QoS architecture for home networks in which QoS provisioning is driven by user preferences and, at the same time, its autonomy reduces direct user participation.

CHAPTER 3

MODA

Contents

3.1 Introduction

FRom previous chapter 2, we have realized that QoS solutions offer mechanisms which affect network parameters like delay, packet loss, variation of delay, or bandwidth assignation. Traffic priority parameters were considered in some QoS solutions at network layer like the Type of Service (TOS) in IP protocol [Pos81] that in practice, it was never deployed. Later, TOS became the Differentiated Services Code Point (DSCP) in DiffServ architecture [BBC+98]. At link layer, traffic priorities are actually predefined traffic classes. Indeed, these QoS solutions work in an isolated manner: it means that QoS solutions at network layer are not aware of QoS solutions at link layer. This is not a restriction but certainly it is better to provide QoS at two or more layers in a coordinated manner. UPnP QoS architecture has provided static mapping of traffic priorities from network layer to link layer. However, it is important to say that these so called user priorities mean actually traffic priorities and there is no user participation in the definition of such priorities. Same QoS mapping issues can be observed at application and transport layers.

We propose a QoS solution that is able to take into account user participation. In this QoS solution, we believe that is mandatory that the different actors have the same understanding about QoS preferences, requirements, and parameters. The use of semantics is the first step toward the development of such QoS solutions. Semantics facilitates knowledge sharing and enables QoS solutions to orchestrate what users want, what applications need, and what network providers can offer. Particularly, for QoS management in the context of home networks, knowledge sharing can allow network providers to understand end-user preferences about the applications as well as QoS requirements of applications.

This chapter has two main sections. The first section presents a state of the art of the use of models (and ontologies) as a language to achieve the representation of knowledge and also to drive the design and development of information systems. The second part of the chapter presents an ontology-based framework named MODA, which stands for Multimedia Ontology Driven Architecture, aimed at allowing knowledge sharing between the different actors in home network scenarios. The main goal of MODA is to provide a semantic common space of knowledge between all the actors in the system. MODA also helps to build QoS policies which are driven by the service user, the service provider, application programmer or all of them at the same time. Consequently, semantic information can be used within decision models in order to provide optimal solutions as in a QoS manager entity assigning and provisioning network resources.

3.2 Model Driven Engineering

The objective of the Model Driven Engineering (MDE) approach is to drive the software development based on domain models rather than implementation or computing models. In other words, MDE focuses on the specification of a system rather than its implementation. Thus, it is a question of making an abstraction of the programming language using an abstract modeling process focused on the use of several standards such as Meta Object Facility (MOF) [Met06], Object Constraint Language (OCL) [Obj10], and Unified Modeling Language (UML) [Uni10] .

Model Driven Architecture (MDA) is a field of the MDE. MDA specifies a methodology for the development of software systems by separating business and application logic from underlying platform technology [OMG03]. MDA specifies three levels in its architecture:

- Representation of Computation Independent Model (CIM) from a business model. CIM describes the context in which the systems will be used.
- Representation of Platform Independent Model (PIM) from CIM. It describes the system itself without any details of its use or its platform. A PIM will be suited for one or several real architectural platforms.
- Representation of Platform Specific Model (PSM) from PIM. At this level, the environment of implementation platforms and languages are known.

MDA allows designing a workflow based on different mappings from CIM to PIM and from PIM to PSM. The mappings can be automated, particularly when a specific model level specifies the mapping rules in a meta-model. In other words, MDA methodology increases the interoperability in heterogeneous environments and provides a method for system integration proposing several model abstraction layers and using generic mapping engines and procedures.

3.2.1 Ontology-Driven Architecture (ODA)

In recent years, the W3C Semantic and Web Best Practices and Deployment Working Group (SWBPD) has proposed a natural extension to the MDA methodology that use semantic models or ontologies, in order to take advantage of Semantic Web technologies. This extension has been defined as the Ontology Driven Architecture (ODA) [TPO+06].

ODA is aimed at extending MDA by providing representation of unambiguous domain vocabularies (e.g. requirements, constraints, services, properties, etc.), model consistency checking and validation as well as to enable new automatic software engineering capabilities. Several works have been carried to illustrate how the ODA approach can be used in the design, development and management of distributed systems. For instance, in [Wag08] ontologies are used to represent knowledge about platform diversity and how this information is used to perform safe configuration of refinement transformation between the platform models.

Indeed, there have been several efforts to integrate the advantages of Semantic Web technologies in the MDA methodology. ODA is one of these efforts and SWBPD proposes some ideas on how Semantic Web technologies can be used in ODA:

- Systems and Software Engineering. In this perspective, the proposed idea is to utilize the Semantic Web as a *classification* or as a *mechanism*. As a *classification*, it groups together tools and techniques for modeling rigorous semantics during specification and design stages of the software life-cycle. As a *mechanism*, it is used to describe, identify, discover and share artifacts among systems and design teams both during design and at runtime.

- Formal Model Specifications. This perspective proposes the use of ontologies and semantic web technologies as a means of formal communication between agents, human or not, participating in the software development process. Indeed, the use of the ontologies as formal or semi-formal model specifications facilitates the communication and knowledge management. Finally, ontologies characteristics like semantics richness, unambiguity, and standard computer readable representation make of ontology driven approaches a natural point of integration between Software Engineering and the Semantic Web.

- Software Lifecycle Support. This perspective proposes the use of ontologies, name spaces, and metadata as providers of language, terminology, and rule standard for a specific domain during software lifecycle.

- Reusable Content and Use of Metadata as Relational Data. This perspective proposes the use of metadata and ontologies to create relational data schemes and models. The use of metadata and ontologies is proposed as powerful descriptors of services, components, and composites in order to facilitate the discovery of services based on precise descriptions. Indeed, Semantic Web techniques facilitate to discover and to use services at different stages of the software Lifecycle.

According to this classification, the research work we have done in this thesis mainly falls in the use of Semantic Web in Systems and Software Engineering.

Certainly, the use of ontologies in a combined ODA-MDA approach allows to obtain important benefits, for instance: system components management at runtime; formal logic-based semantic for description of components; reasoning and querying on ontologies allowing for verification; as well as, checking the consistency of the system configuration whether this is done during development or run-time.

Another important effort of the OMG in order to integrate semantics and MDA together through the use of ontologies is the Ontology Definition Metamodel (ODM). The following subsection explains in more detail this approach.

3.2.2 Ontology Definition Metamodel (ODM)

Recently, OMG has proposed the Ontology Definition Metamodel (ODM) [Ont09] specification in order to define the basis for representation, management, interoperability, and application of business semantics. ODM gives a stronger support to MDA by providing a family of independent metamodels, profiles, and mappings among the metamodels corresponding to international standards for ontology definition. We can say that MDA plays the role of methodology but ODM provides a path to effectively apply MDA for ontology-based applications, that is, using ontology-based models to produce platform specific models.

ODM makes clear that by using ontologies, additionally to UML 2.0, it is possible to have a reliable set semantics and model theory, which allow automated inference on models. Currently, the use of reasoners on UML models is not possible because it lacks a formal model theoretic semantics. In contrast, OWL DL implements a subset of first order logic, which enables the use of automated inferences (reasoning).

ODM specification presents two perspectives in order to characterize the domain of applications to which the ODM is intended to support: Model-centric and Application-centric perspectives.

Model-centric perspective is associated with conceptualizations. It characterize the ontologies themselves and it is concerned with the structure, formalism and dynamics of the ontologies. Model-Centric perspectives are:

- Level of authoritativeness. It refers to the level of authority of the ontology in the domain. The level of authoritativeness is directly related to the person or organization who develop the ontology, that is, the ontology may be *highly authoritative* if it is defined by the organization that is responsible for specifying the conceptualization. On the other side, the ontology *may not be very authoritative* if it was specified by someone that is distant from the responsible organization of the conceptualization.

- Source of structure. From this perspective, an ontology is called *transcendent* when it is defined externally to the applications that use it. Changes to a transcendent ontology are made through a revision process. On the other hand, *immanent* ontologies have structures that may be defined by applications based on content knowledge or inferences.

- Degree of formality. It refers to the level of formality used to specify the conceptualization. Formality goes from *highly informal* ontologies where there are no formal axiom expressions, to *highly formal* ontologies containing formal axioms.

- Model Dynamics. It refers to the frequency at which an ontology structure evolves. A *stable* or *read-only* dynamics mean that the ontology structure changes rarely. A *volatile* dynamics refers to ontology structures that change often.

- Instance Dynamics. Like model dynamics, instance knowledge in an ontology may be *stable* (read-only) or *volatile*, based on the frequency of changes to the instance knowledge. If the instance knowledge of ontology changes often then the instance dynamics is *volatile*, and *stable* when it does not.

Application-centric perspectives are related with the way applications utilize and manipulate ontologies. Application centric perspectives are:

- Control/Degree of manageability. This perspective indicates who decides when and what changes to make to an ontology. The control of manageability can be *internally focused* when the responsible for the ontology is the only entity who decides the changes to be applied; and also, it can be *externally focused* when changes to the ontology are decided by external (not ontology responsible) entities.

- Application changeability. It refers a kind of application, *static* or *dynamic*, that use an ontology. *Static* applications are developed once and may be periodically updated. *Dynamic* applications may be constructed at run time.

- Coupling. This perspective describes the degree of sharing ontologies by applications. It can be *tightly coupled* when applications are closely coupled since they must interoperate at run time when they commit to an ontology. On the other hand, applications are *loosely coupled* because they share an ontology but not at run time.

- Integration focus. The integration focus can be application and information integration. *Application integration* describes the perspective where ontologies are used to link programs, that is, ontologies are used to specify the structure of interoperation between applications. *Information integration* rather includes ontologies that describe the content structure allowing applications exchange information about shared objects.

- Lifecycle usage. This refers to the phase of software life cycle at which the ontology is used. Thus, ontologies can be used, for example, at *designing time* or *run time*.

Taking into account the before mentioned perspectives, the ODM specification proposes application areas that share perspectives values. Such application areas are illustrated in Table 3.1:

- Business Applications. They are characterized by a transcendent source of structure, a high degree of formality and external control. Within business applications, we can find run-time interoperation, application generation and ontology lifecycle.

- Analytic applications. They have highly changeable and flexible ontologies, using large collections of mostly read-only instance data. Here we can subclassify applications in emergent property discovery and exchange of complex data sets.

- Engineering applications. They have transcendent source of structure, ontologies of theses applications are more authoritative and they are controlled internally by users. Within these kind of applications we can find information system development and ontology engineering.

We think that for our research work, the application area of such semantic framework is the *engineering applications*. We explained this affirmation from the two perspectives proposed by the ODM specification:

From a ***model-centric*** perspective, we need a semantic framework that should be generic enough to be used by the different actors in our system, i.e., home-users, applications programmers, and home network providers.

Application Clusters		Characteristic Perspective Values									
		Model Centric					Application Centric				
	Description	Authoritative	Structure	Formality	Model Dynamics	Instance Dynamics	Control	Change	Coupling	Focus	Life Cycle
Business Applications			Transcendent	Formal			External				
	Run-time interoperation	Least/Broad	Transcendent	Formal	Read-Only	Volatile	External	Static	Tight	Information	Real-time
	Application Generation	Most/Deep	Transcendent	Formal	Read-Only	Read-Only	External	Static	Loose	Application	All
	Ontology lifecycle	Middle/Broad & Deep	Transcendent	Semi-Formal & Formal	Read-Only	Read-Only	External	Static	Tight	Information	Real time
Analytic Applications					Volatile	Read-only		Dynamic	Flexible		
	Emergent Property Discovery	Broad & Deep	Immanent	Informal	Volatile	Read-Only	Internal & External	Dynamic	Flexible	Information	Real time
	Exchange of complex data sets	Broad & Deep	Immanent	Informal	Volatile	Read-Only/Volatile	Internal & External	Dynamic	Flexible	Information	Real time
Engineering Applications		Broad & Deep	Transcendent				Internal				
	Information System Development	Broad & Deep	Transcendent	Semi-Formal/Formal	Read-Only	Volatile	Internal	Evolving	Tight	Information	Design Time
	Ontology Engineering	Broad & Deep	Transcendent	Semi-Formal/Formal	Volatile	Volatile	Internal	Evolving	Flexible		Design Time

Table 3.1: Application Clusters: Perspective Values

Certainly, the ontology-based models should be also generic enough to characterize the points of view of the different actors as well as the different possible home scenarios. At the same time, we think that we need a semantic framework that contains generic and wide accepted ontology-based models. Thus, such ontologies should be based on accepted standards and specifications. The objective is to have semantic models that are *highly authoritative* as possible. As a consequence, we think that changes to the ontologies should be made through a revision process, i.e., we need a semantic framework that includes *transcendent* ontologies.

Considering that the ontologies should be based on accepted standards, their model dynamics should be *stable*, it means that the ontology structure change rarely as well as its instances (*instance dynamics*). Also, we think that the semantic framework should be *formal* enough to conceptualize the domain of knowledge and its facts in order to increase inference of new knowledge by using inference engines e.g., reasoners.

From an ***application-centric*** perspective, we think that changes to the ontologies should be made by the semantic framework responsible and not by an external entity, i.e., the control of manageability should be *internally focused*.

We envisage that the applications will use the semantic framework as source of knowledge, thus they are *static* applications that can be periodically updated and not constructed at run-time. However, the applications that use the semantic framework will interoperate because the information shared by ontologies includes data from the different actors, i.e., the applications' *degree of coupling* should be *tightly coupled*.

Indeed, applications will share the information contained in the semantic framework and also the ontologies will describe the content structure allowing such exchange of information (*information integration*).

Clearly, applications and actors may use the semantic framework at different times of the lifecycle usage, for example, application programmers will use it at designing time, home-users will use it directly when defining its preferences and indirectly when the session is configured based on the selected applications and consumed contents.

3.2.3 Summary

This section has presented the efforts on modeling, programming and deployment of software systems driven by models (and meta-models). For this purpose, it has been presented how MDA specifies a methodology starting from high-level representations of a Computing Independent Model (CIM) to Platform Specific Models (PSM) passing through the representation of Platform Independent Model (PIM).

Ontology Driven Architecture (ODA) has been proposed as an effort to integrate the advantages of Semantic Web technologies with the MDA methodology. ODA proposes the use of ontologies as models to drive the design of architectures, development and deployment of systems. Indeed, the use of ontologies allows the representation of unambiguous domain vocabularies in order to improve the semantics of existing modeling languages like UML.

Ontology Definition Metamodel (ODM) enforces the use of ontologies to drive system architectures. While MDA provides the methodology, ODM provides a family of metamodels, profiles, and mappings among the metamodels for ontology definition. ODM proposes the use of ontologies additionally to UML, in order to have a more complete semantics, then taking advantages of automated inference on models.

Also, this section has proposed the characteristics that a semantic framework should possess in order to allow the actors (home users, application programmers, and home-network providers) to have a common space, where they can share the semantics of their domain of knowledge. Such characteristics are defined by the ODM specification. We think that the application area of the required semantic framework corresponds to *engineering applications*. Certainly, we are looking for a semantic framework that includes models with structures that do not change frequently and if they change, a revision process is necessary. Besides, such changes are managed by the semantic framework responsible and not by external entities.

As we can see, currently there are important research work that proposes the use of models to establish the basis for representation, management, interoperability, reusability, and application of semantics in software engineering of systems and architectures. We have already seen that ODM encourages the utilization of ontologies in the Model-Driven Architecture. The next section is focused on knowledge representation with ontologies.

3.3 Knowledge Representation

Knowledge Representation (KR) is a subdomain of the Artificial Intelligence (AI) field of computer science. Knowledge Representation is a fundamental area of AI because KR techniques allow the knowledge of a domain to be stored in an computer readable format. The stored knowledge is used by automated reasoning processes in order to infer more information or to make new conclusions in an automated manner [RNC+09]. In [DSS93], the authors defined the KR concept from the five different roles it plays:

- KR is a substitute of the thing being represented, and it is used to reason about it.

- KR is a set of ontological commitments. KR is inherently linked to the person that provides such representation of the knowledge. The person who develops the KR establish certain commitments in order to represent and to notice certain aspects of what is being represented.
- KR is a fragmentary theory of intelligent reasoning. The initial conception of a representation indicates how people reason intelligently. It is fragmentary because KR incorporates only part of the belief that motivated the representation; and also, the belief is, in turn, only a part of the phenomenon of intelligent reasoning.
- KR is a medium for pragmatically efficient computation. A KR made of a well organized information facilitates the computation of new conclusions or inferences.
- KR is a language through which humans can communicate. Indeed, KR is a medium of expression and communication between humans, humans and computers, or between computers.

From the above enlisted roles of KR, one can see that there is a link between KR and the language used for the representation, the phenomenon of reasoning and automated reasoning, and the semantics. Indeed, there is a correspondence between the part of the world being represented and its representation, such correspondence is the semantics of the representation. KR captures the *meaning* of concepts, properties, and relationships being represented. KR uses a language in order to express the meaning of things. In part, a language is representational because it carries the meaning of the representation [BL85].

Among the first KR languages, one can find *Frame Systems* [Min74] and *Semantic Networks*[Qui67]. Frame Systems propose the use of a data structure called *frame* to represent information. Frames are related via *slots*. Semantic networks are based on directed graphs where nodes represent objects or concepts and arcs represent relations between these objects. Both, Frame systems and Semantic Networks are structural representations of a domain of knowledge. Semantics appears while defining the structure of a directed graph [Hay79]. However, it is important to say that semantic networks and frames are non-logical approaches to KR.

Formal languages allow for formal descriptions or representations. Looking for formal and logic-based KR languages, the Description Logics (DL) research area was developed after semantic networks and frames. DL is based on first order logic and, thus, it is possible to represent other kind of relationships than just inheritance (IS-A) relations. Thus, by using a DL language, one gain in formal expressiveness. However, as argued in [BL84], there is a tradeoff between the expressiveness of a representation language and the difficulty of reasoning over its representations: the more expressive is the language, the harder is the reasoning.

DL languages can be used to characterize a domain of knowledge. Such characterization can be made through the definition of an *ontology* of the domain of knowledge. The following subsection gives more details about ontologies.

3.3.1 Ontologies and Semantics

The ontology term comes from philosophy, where ontology is defined as "a science or study of being: specifically, a branch of metaphysics relating to the nature and relations of being" [MW11]. One issue with the word *ontology* is that it does not have a universal definition and, as a matter of fact, it may have several interpretations. In other words, ontology definitions may vary according to the domain where they are used. In [GG95], Guarina and Giaretta present a deep analysis about the definition and interpretation of the ontology term.

Particularly, in the domain of computer science, one of the most known definitions is given by [Gru93]: "An ontology is an explicit specification of a conceptualization", where conceptualization is the base of a body of formally represented knowledge: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them [GN87].

Ontologies use and propose a vocabulary in order to characterize the domain of knowledge being conceptualized. This vocabulary provides a set of words in order to describe the facts in the domain of knowledge [CJB99]. The ontology's vocabulary also allows defining axioms. Axioms are logical statements that are assumed to be true. Axioms help to represent more information about the concepts and properties in the ontology.

Ontologies also help to organize the information and knowledge of the domain being represented. Classification of concepts according to a given criterion, and creation of taxonomies following inherent properties of the concepts are part of the main activities to build an ontology. Research work that proposes methodologies to build ontologies, recommend concept classification and taxonomy creation as necessary steps in the ontology creation process [NM01, GPFDV96].

Ontologies are developed with different goals in several areas. For example, in order to cope the lack of sharing understanding of a given domain of knowledge. [UG96] identifies three main uses of ontologies: for communication between people, interoperability among systems, and to obtain system engineering advantages like re-usability, reliability and specification. Indeed, ontologies enable the actors who agree on such knowledge representation to have the same understanding about the domain of knowledge.

W3C has proposed the use of ontologies as explicit conceptual models in Ontology Driven Architectures (ODA) in the context of Model Driven Architectures (MDA). Indeed, ontologies allow defining concepts and properties in an unambiguous manner, thus they facilitate users to have the same understanding of what has been modeled.

Ontologies can be developed using different kind of vocabularies, from highly informal to rigorously formal languages [UG96]. Looking for the automation of ontologies, there have been several efforts to create languages to develop ontologies like e.g. Knowledge Interchange Format (KIF) [GF92] and the OWL ontology web language [BVHH$^+$04] . Nowadays, OWL is one of the most popular languages used to develop ontologies. The following section gives a description of OWL.

3.3.2 Introduction to OWL: Web Ontology Language

OWL stands for Web Ontology Language (not WOL for esthetic reasons) is written using XML and built on top of Resource Description Framework (RDF) and RDF Schema (RDFS) . The use of XML to write OWL ontologies allows the interchange of ontologies between different systems and platforms. Also, RDF establishes the basis for semantic expressions using triples composed by a subject, a predicate, and an object. In Figure 3.1, the predicate or property represents the relationship that exists between things, which are represented by the subject and object.

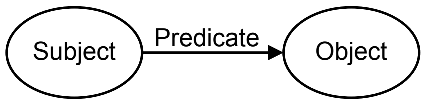

Figure 3.1: An RDF Triple.

OWL is a vocabulary extension of RDF. OWL ontologies allow representing the meaning of terms in vocabularies and the relationships between those terms. By extending RDF vocabulary along with a formal semantics, OWL has more facilities for expressing meaning and semantics than XML and RDF to represent machine interpretable content.

Even when OWL was designed for semantic web applications, its utilization has been spread to any area where ontologies are used. OWL is intended to allow applications to process information instead of only present it to the user. OWL comes in three sublanguage versions: OWL lite, OWL DL, and OWL full. The difference between them is the level of expression they can reach [MVH04].

- OWL lite: it is the simplest sublanguage version of OWL. It provides a minimal subset of language features. OWL Lite provides the basics for subclass hierarchy construction: subclasses and property restrictions; but properties can be made optional or required. Implementations that support only OWL Lite are not able to perform reasoning tasks.

- OWL Description Logics (DL): it supports those users who want the maximum expressiveness while retaining computational completeness done in finite time. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions, e.g. a class cannot be an instance of another class. OWL DL has a correspondence with description logics, a field of research that has studied the logics that forms the formal foundation of OWL.

- OWL Full: it is meant for users who need maximum expressiveness and the syntactic of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

OWL-DL is the most popular language version of OWL. This is because it has a rich expressiveness but also it supports reasoning tasks like subsumption, equivalence, consistency, and instantiation checking.

Figure 3.2 shows the structure of OWL ontologies (OWL version 2) [MPSP09] using UML. In this section, we only explain such OWL structure. The following section will present the corresponding definitions illustrated in this figure. As we can see in Figure 3.2, ontologies are composed of a set of axioms and annotations. Axioms are statements that say what is true in the domain. The axioms can have annotations whose values can be constants or owl entities. Annotations are used to associate information with an ontology, for example, the reason of a concept's name or the ontology's author.

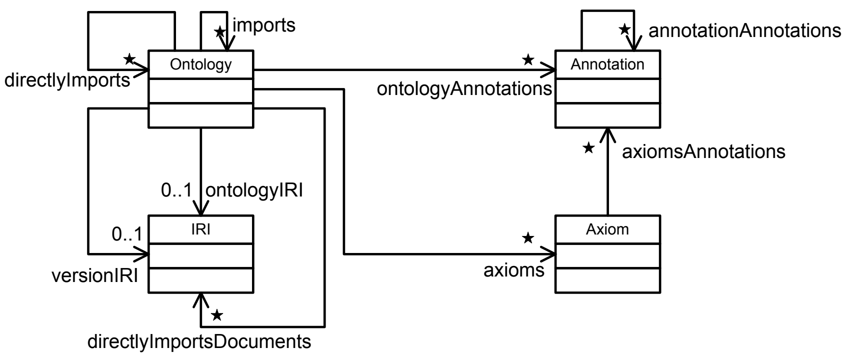

Figure 3.2: Structure of OWL ontologies.

Ontologies can *import* other ontologies, thus enabling ontology (and knowledge) re-usability between ontologies. By importing an ontology, the ontology who imports has only-read access to all elements defined in the imported ontology. The imported ontology will suffer no changes in its content, but the ontology that imports can add new elements (axioms, subclasses, individuals, relations, etc.) Figure 3.3 illustrates the hierarchy of OWL entities, literals and anonymous indi-

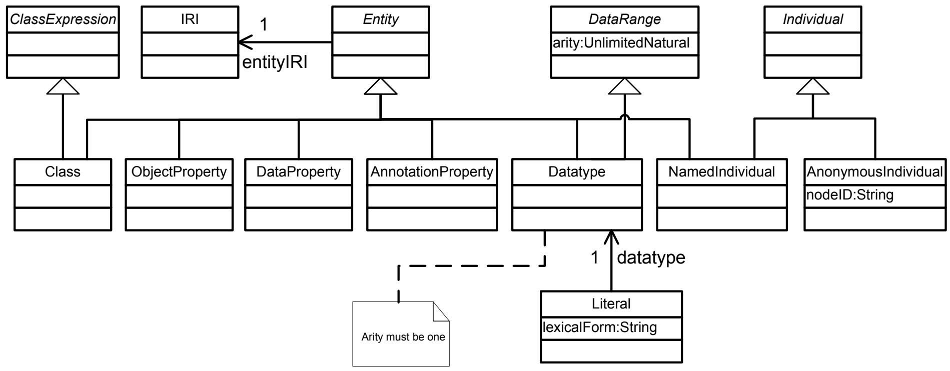

Figure 3.3: Entities in OWL ontologies.

viduals (OWL version 2). This Figure depicts that an *entity* has associated a unique Internationalized Resource Identifier (IRI) . The following classes have a heritage relationship from the class *entity*:

- Class: Classes can be understood as sets of individuals.
- ObjectProperty: Object properties connect pairs of individuals.
- DataProperty: Data properties connect individuals with literals. In some knowledge representation systems, functional data properties are called attributes
- AnnotationProperty: Annotation properties can be used to provide an annotation for an ontology, axiom, or an IRI.
- DataType: Datatypes are entities that refer to sets of data values. They contain data values such as strings and numbers.
- NamedIndividual: Individual that has a defined (given) name and is identified using an IRI.

As we can see, *Class* inherits also from *ClassExpression* which are the classes created from the statements of description logics. We find that *NamedIndividuals* also inherit from the *Individual* class, which also is the generalization of the *AnonymousIndividual* class.

3.3.2.1 OWL-DL definitions

This subsection presents the definitions of the concepts used in OWL ontologies. These definitions are the basis to better understand ontology-based models created with OWL-DL.

Description languages are necessary to represent the knowledge of a domain. The language $\mathcal{AL}$ (attributive language) has been introduced in [SSS91] as a minimal DL language. Description languages are distinguished by the constructors they provide.

Nowadays, there is a family of $\mathcal{AL}$-languages in which each language provides certain constructors and properties that define the level of expressiveness of the language. For example, we can add concept constructors (e.g., union or intersection of two or more atomic concepts), role constructors (e.g., union or intersection of two or more atomic roles), or restrictions on role interpretations (e.g. cardinality restrictions). OWL-DL belongs to the family of $\mathcal{AL}$-languages.

The basic description language $\mathcal{AL}$ allows concept descriptions defined by the following syntax rule:

DL Syntax			Description
C,D	$\rightarrow$	A|	(Atomic concept)
		$\top$|	(universal concept, top concept)
		$\bot$|	(bottom concept)
		$\neg$A|	(atomic negation)
		C$\sqcap$D|	(intersection)
		$\forall$R.C|	(value restriction)
		$\exists$R.$\top$	(limited existential quantification).

Where the standard notation uses the letters *A* and *B* to represent atomic concepts, the letters *C* and *D* to denote concept descriptions. Atomic roles or properties are denoted by the letters *R* and *S*.

In order to define a formal semantics of $\mathcal{AL}$-concepts, it is defined an *interpretation* $\mathcal{I}$ as follows [BCM$^+$03]:

Definition 3.1 ***Interpretation*** $\mathcal{I}$. *An interpretation $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ named the domain of $\mathcal{I}$ and an interpretation function, which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.*

The interpretation function $\mathcal{I}$ can be extended to concept descriptions by the following definitions:

$$\top = \Delta^{\mathcal{I}} \tag{3.1}$$

$$\bot = \emptyset \tag{3.2}$$

$$\neg A^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \tag{3.3}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \tag{3.4}$$

$$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \tag{3.5}$$

$$(\exists R.\top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \exists b.(a,b) \in R^{\mathcal{I}}\} \tag{3.6}$$

Where letters *a*, *b* represent individuals. OWL-DL is based on DL $\mathcal{AL}$. OWL-DL can express role hierarchies as follows:

$$R \subseteq S \equiv \{a \in \Delta^{\mathcal{I}} | \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow (a,b) \in S^{\mathcal{I}}\} \tag{3.7}$$

Where the letters *R* and *S* denote atomic roles; the letters *a* and *b* denote individuals. OWL-DL provides nominal concept constructors i.e., concepts that have exactly one single instance. By using nominal concept constructors one can represent the "one-of" construct. Its semantics is defined as follows:

$$I \equiv I^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} with |I^{\mathcal{I}}| = 1 \tag{3.8}$$

OWL-DL also provides the use of *inverse* roles i.e., for modeling relationships between objects that apply in both senses. The semantics of inverse roles is defined as follows:

$$R^{-} \equiv \{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\} \tag{3.9}$$

OWL-DL offers *role restriction*. It establishes a constraint on the cardinality of the set of role fillers i.e., a restriction on the minimum or maximum number of times an instance of an entity may participate via a given role in instances of the relationship. The semantics of the unqualified number restriction of a role is defined as follows:

$$\geq nR \equiv \{a \in \Delta^{\mathcal{I}} || \{b \in \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \geq n\} \text{ (minimum)} \tag{3.10}$$

$$\leq nR \equiv \{a \in \Delta^{\mathcal{I}} || \{b \in \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \leq n\} \text{ (maximum)} \tag{3.11}$$

$$= nR \equiv \{a \in \Delta^{\mathcal{I}} || \{b \in \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| = n\ \} \text{ (exact)} \tag{3.12}$$

OWL-DL uses datatype properties. OWL datatypes come from the XML Schema type system[1].

Now that the constructors of OWL-DL have been presented, in the following paragraphs we present the definitions of the main elements and concepts used in OWL-DL ontologies for a given interpretation $\mathcal{I}$.

[1]http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#typesystem

Definition 3.2 ***Domain of Knowledge***. *It is what people know about something(s), a part, or an area of the world. The domain of knowledge is what the ontology models through the use classes, individuals, and properties (relationships).*

In ontologies, we find three main concepts: *Classes* (atomic or description concepts), *Properties* (roles), and *Individuals*. They are defined as follows.

Definition 3.3 ***Individual***. *Individuals in OWL are the objects in the domain being modeled. They are also called class instances.*

An individual o has the following DL syntax and semantics definitions:

$$o \equiv o^{\mathcal{I}} \in \Delta^{\mathcal{I}} \tag{3.13}$$

Definition 3.4 ***Class***. *A class is a set of individuals. Classes are described using formal descriptions. Its description states the requirements for an individual to belong to that class.*

Elementary descriptions of classes are called atomic concepts, whereas complex descriptions can be constructed from atomic concepts using concept constructors. Concept constructors like for example: union, intersection, and complement of classes allow describing complex classes. The semantics of an atomic concept (class) C is defined as follows:

$$C \equiv C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \tag{3.14}$$

The disjunction constructor of two classes C and D is expressed as follows:

$$C \sqcup D \equiv C^{\mathcal{I}} \cup D^{\mathcal{I}} \tag{3.15}$$

In the same manner, the conjunction constructor is defined as follows:

$$C \sqcap D \equiv C^{\mathcal{I}} \cap D^{\mathcal{I}} \tag{3.16}$$

The complement of a class C is defined as follows:

$$\neg C \equiv \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}} \tag{3.17}$$

There is an inherent taxonomy of classes in an ontology. The taxonomy includes sub-classes and super-classes relationships. Within ontologies, all the members of a class are *subsumed* (included) by its super-classes. Also, OWL allows for multiple inheritance, in other words, an individual belongs to more than one class. It is important to remark that multiple inheritance may lead to inconsistencies in the ontology when it is not well defined. In these cases a reasoner helps to identify such errors. For two classes C and D, the following semantics description of subsumption is provided:

$$C \sqsubseteq D \equiv C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \tag{3.18}$$

Definition 3.5 ***Property***. *Property is the binary relation between objects or individuals. In other words, a property links two individuals of the ontology. In OWL-DL, there are three types of properties: Object, Datatype, and Annotation Properties. A property has a Domain, from where the property comes, and a Range, where the property goes to.*

A property can be subsumed by its super-property. In this case, the property is a sub-property of its super-property. For a sub-property, its domain and range are subsumed by the domain and range of its super-property respectively. It is important to say that we should speak of "instance of property" instead of just "property", because property instances actually establish the link between objects (individuals of classes). For the sake of the language, the ontological community uses just the term "property" and not "instance of property".

Definition 3.6 ***Object Property***. *Object Property P is the binary relation between two individuals or objects.*

The syntax and semantics of an object property P are defined as follows:

$$P \equiv P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \tag{3.19}$$

Definition 3.7 ***Datatype Property***. *A Datatype Property U links an individual to a datatype values (integer, string, float, etc).*

A datatype property has the following DL syntax and semantics:

$$U \equiv U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}_{D} \tag{3.20}$$

Definition 3.8 ***Annotation Property***. *An Annotation Property is meta-data associated to classes, properties or individuals. It allows to provide more information about what is being modeled.*

Definition 3.9 ***Property's Domain***. *The Domain of a property P is the class or classes from where the property comes.*

The semantics definition of the Domain of a property P is:

$$\geq 1P \sqsubseteq C_i \equiv P^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}_{\mathcal{D}} \tag{3.21}$$

Definition 3.10 ***Property's Range***. *The Range of a property is the class or classes to where the property establishes the relation.*

The semantics definition of the Range of a property P is:

$$\top \sqsubseteq \forall P.C_i \equiv P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}} \tag{3.22}$$

OWL allows to define the inverse property of an object property. We have already defined inverse properties (see definition 3.9). In the context of OWL, we can provide the following interpretation:

Definition 3.11 ***Inverse Property***. *When an object property P is defined, P links object A to object B (in this direction). The inverse property P' of P is the relation from object B to object A (in this direction).*

The semantic definition of an inverse property P was given in the expression 3.9.

In OWL, object properties may have characteristics. Such characteristics allow the ontology expert or modeler to better define the relationships between objects (individuals) in the domain of knowledge. Object property characteristics contribute to unambiguous definitions in the ontology. The object property characteristics are: Functional, Inverse Functional, Symmetric, Asymmetric, Transitive, Reflexive, and Irreflexive. They are defined as follows:

Definition 3.12 ***Property's Functional Characteristic***. *A property P is functional if it links an object A to an object B, and only object B. If, for some reason, P is also linking the object A to another object C, then we can infer that objects B and C must be the same individual because P is functional. In the latter case, an important remark is that if object B and C were explicitly defined as two different objects then the ontology is not consistent. Property's functional characteristic also applies to Datatype properties.*

The semantic definition of a functional property can be stated as follows: For an *interpretation* $\mathcal{I}$, a property P is functional if and only if

$$\{(A, B), (A, C)\} \subseteq P^{\mathcal{I}} \text{ implies } B = C \tag{3.23}$$

Definition 3.13 ***Property's Inverse Functional Characteristic***. *A property P is inverse functional if it links an object A and only A, to an object B. If, for some reason, P is also linking an object C to the object B, then the objects A and C must be the same because P is inverse functional.*

The semantic definition of an inverse functional property is as follows: For an *interpretation* $\mathcal{I}$, a property P is inverse functional if and only if

$$\{(A, B), (C, B)\} \subseteq P^{\mathcal{I}} \text{ implies } A = C \tag{3.24}$$

Definition 3.14 ***Property's Symmetric Characteristic***. *A property P is symmetric if P links object A to object B (in this direction) and also P relates object B to object A (in this direction).*

This definition can be stated as follows: A property P is symmetric if and only if

$$\{(A, B), (B, A)\} \subseteq P^{\mathcal{I}} \tag{3.25}$$

Definition 3.15 ***Property's Asymmetric Characteristic***. *A property P is asymmetric if P links object A to object B but P cannot link object B to object A.*

In other words, a property P is asymmetric if and only if

$$\{(A, B), (B, A)\} \nsubseteq P^{\mathcal{I}} \tag{3.26}$$

Definition 3.16 ***Property's Transitive Characteristic***. *A property P is transitive if P relates object A to object B, object B to object C, and also object A to object C.*

This definition means that if a property P is transitive, then for any objects A, B, and C the following statement is true:

$$\{(A, B), (B, C)\} \subseteq P^{\mathcal{I}} \text{ implies } \{(A, C)\} \subseteq P^{\mathcal{I}} \tag{3.27}$$

Definition 3.17 ***Property's Reflexive Characteristic***. *A property P is reflexive if P links an object A to the same object A. Clearly, P can link object A to others objects.*

Definition 3.18 ***Property's Irreflexive Characteristic***. *A property P is irreflexive if P links object A to object B but A and B are never the same object.*

In OWL, it is possible to define classes in two different ways. A *named class* is the easiest way to create a class and it consists in just giving a name to the class and nothing more. *Anonymous class* creation is based on the descriptions we make about the class. These descriptions are actually declared restrictions on object and datatype properties of a class. Also, it is possible to create a named class and then define a set of restrictions on the class properties. In this case, we are also creating an anonymous super-class of the named class. The different types of restrictions are classified in three categories: Quantifier, Cardinality, and *hasValue* restrictions. Quantifier restrictions are subclassified in Existential, Universal. They are defined as follows:

Definition 3.19 ***Existential Restriction***. *An existential restriction states that a property P must have at least one instance of P linking to objects that belong to a specific Range C. In other words, by defining an existential restriction on a property P, we are describing the classes that contain individuals that have at least one relationship using P to individuals in the specific Range C.*

The semantics of existential restriction is interpreted as follows:

$$\exists P.C \equiv \{A \in \Delta^{\mathcal{I}} \mid \exists B.(A, B) \in P^{\mathcal{I}} \wedge B \in C^{\mathcal{I}}\} \tag{3.28}$$

Definition 3.20 ***Universal Restriction***. *A universal restriction states that all instances of a property P must link to objects in a specific Range C. In other words, a universal restriction on property P describes classes of individuals that only have relationships to a specific Range C along the property P.*

The semantics of universal restrictions is interpreted as follows:

$$\forall P.C \equiv \{A \in \Delta^{\mathcal{I}} \mid \forall B.(A, B) \in P^{\mathcal{I}} \rightarrow B \in C^{\mathcal{I}}\} \tag{3.29}$$

Cardinality restrictions define constraints on the number of relationships that an object can participate in for a given property P. There are three types of cardinality restrictions: *Exact, Minimum, and Maximum Cardinality Restrictions*. As their names suggest, exact cardinality indicates the exact number of relationships using property P that an object can participate in. Minimum cardinality specifies the minimum number of relationships that an object can participate in using property P. Maximum cardinality specifies the maximum number of relationships that an object can participate.

The definitions for unqualified cardinality restrictions of an object property have been provided in the expressions 3.12, 3.11, and 3.10. For qualified cardinality restrictions, the following definitions are provided.

Given an Interpretation $\mathcal{I}$, the minimum cardinality for an object property P is defined as follows:

$$\geq nP.C \equiv \{A \in \Delta^{\mathcal{I}} || \{B \in \Delta^{\mathcal{I}} | (A, B) \in P^{\mathcal{I}} \wedge B \in C^{\mathcal{I}}\}| \geq n\} \tag{3.30}$$

The maximum cardinality for an object property P has the following definition:

$$\leq nP.C \equiv \{A \in \Delta^{\mathcal{I}} || \{B \in \Delta^{\mathcal{I}} | (A, B) \in P^{\mathcal{I}} \wedge B \in C^{\mathcal{I}}\}| \leq n\} \tag{3.31}$$

The exact cardinality for an object property P has the following definition:

$$= nP.C \equiv \{A \in \Delta^{\mathcal{I}} || \{B \in \Delta^{\mathcal{I}} | (A, B) \in P^{\mathcal{I}} \wedge B \in C^{\mathcal{I}}\}| = n\} \tag{3.32}$$

Definition 3.21 ***hasValue Restriction***. *A hasValue restriction on a property P means that P must have at least one instance linking to a specific object A. In other words, a hasValue restriction on property P describes classes of individuals that must have at least one relationship to a specific object A.*

It is important to say that *hasValue* restrictions are semantically equivalent to an existential restriction along the same property P, with a filler that is an enumerated class that contains the object A, and only the object A, used in hasValue restriction. The semantics definition of a *hasValue* restriction is:

$$P : o \equiv \{x \mid (x, o^{\mathcal{I}}) \in P^{\mathcal{I}}\} \tag{3.33}$$

OWL is based on the Open World Assumption (OWA). OWA means that we cannot assume that something does not exist unless it is explicitly stated that it does not exist. As a consequence, we cannot say that something is false just because it was not stated as truth in the ontology. Ignoring the OWA can lead ontology developers to inconsistencies or apparent error inferences in the ontology. There are some concepts in OWL that help to better define OWL ontologies in order to take into account OWA. Such concepts are defined as follows:

Definition 3.22 ***Disjoint Classes***. *This concept applies at the level of classes. If two or more classes are defined as disjoint classes, the individuals in each class cannot be members of more than one class of the disjoint classes. If disjoint classes are not defined, OWL classes are assumed to overlap, meaning that their individuals can be members of several classes.*

Semantics of disjointed classes C_i, C_j is defined as follows:

$$C_i \sqcap C_j = \bot, i \neq j \equiv C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} = \emptyset, i \neq j \tag{3.34}$$

Definition 3.23 ***Closure Axiom***. *Closures axioms act at properties. A closure axiom on a property P is the definition of an existential and universal restriction at the same time on P. As result, a closure axiom on P describes the class of individuals that have at least one relationship through P to objects in a specific range R and only to R. Considering that, range R can be a set of classes, the universal restriction on P must be defined as the union of all classes in the Range R.*

3.3.3 Reasoning

One advantage of using OWL-DL as language for knowledge representation is its support for reasoning tasks. We start this subsection by explaining what *reasoning* is in the context of knowledge representation using description logics languages. Reasoning is a “service that allows one to infer implicitly represented knowledge from the knowledge that is explicitly contained in the knowledge base”[BCM^{+}03].

We can see a *reasoner* as a software entity that executes reasoning algorithms on e.g. OWL-DL ontologies in order to infer knowledge non explicitly represented in such OWL-DL ontology. The standard reasoning tasks or procedures that a reasoner can offer are: subsumption, equivalence, consistency, and instantiation checking. Here they are briefly explained:

- Subsumption checking is the basic inference on concept expressions in description logics. Subsumption checking is verifying whether the concept D (the *subsumer*) is considered more general than the one denoted by C (*the subsumee*). In other words, subsumption checks whether concept C is a subset of concept D. (See expression 3.18). Another result of subsumption checking is that the reasoner can propose an inferred hierarchy of classes besides the one that is explicitly stated in the ontology.

- Equivalence checking consists in verifying when two concepts C and D are equivalent. In other words equivalence checks if the represented knowledge is minimally redundant.

- Consistency checking. This task aims at verifying if each concept in the knowledge source allows one instance at least. For instance, when a class definition or description (based on restrictions) does not allow to instantiate it (create an individual of that class), then that class is inconsistent.

- Instantiation checking. This task is executed on the assertions made from the definitions stated in the ontology. In other words, it checks if an individual a is an instance of a class C.

There are two main kinds of algorithms aimed at performing reasoning tasks: *structural subsumption* and *tableau-based*[SSS91]. The first ones are used to compute subsumption of concept, but they are only complete for simple languages with little expressivity. For more complex Description Logics, the *tableau*-based algorithms are used to perform reasoning tasks including extensions to $\mathcal{AL}$-concepts like number restrictions and transitive roles.

Reasoning in OWL is based on the *open world assumption*. Unlike Relational Databases (Close World) where one obtain a negative or false value in return when it cannot find some data; in OWA, one cannot assume that something is false just because it has not been stated to be true. In other words, a reasoner has not complete knowledge, thus it can only infer knowledge from the asserted statements. As a consequence, in the OWA, one assumes that knowledge can always be added later to the ontology.

3.3.4 Rules

Despite the fact that OWL is more expressive than RDF or RDFS, OWL has its limits and there are users that need more expression capabilities particularly for

OWL properties. Semantic Web Rule Language (SWRL) [HPSB+04] is aimed at increasing expression capabilities of ontologies. This subsection gives a description of SWRL.

Semantic Web Rule Language (SWRL) is based on a combination of the OWL DL and OWL Lite languages with the Rule Markup Language (RuleML) [HBG+11] and is intended to extend the set of OWL axioms by including Horn-like rules (implications or if-then conditional statements).

In a human readable syntax, the proposed rules are of the form of an implication between an antecedent (body) and consequent (head). The user can read the rule as follows: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. A SWRL rule can be represented like follows:

$$Antecedent(body) \Rightarrow Consequent(head)$$

Antecedent (body) and consequent (head) are composed of zero or more atoms. An atom can be any unary predicate like when stating that an instance belongs to a class, binary predicate like when using object and data properties, or built-ins. In SWRL, body and head can exist with no atoms. An empty body is always treated as true, so the consequent must also be satisfied by every interpretation; an empty consequent is evaluated to false, so the antecedent must also not be satisfied by any interpretation. Multiple atoms are treated as a conjunction $(a_1 \& a_2 \& ... \& a_n)$. SWRL integrates the use of Built-ins. SWRL built-ins are also aimed at increasing the expressivity of ontologies. They cover several aspects, thus there are built-ins for:

- Comparisons
- Mathematical operations
- Strings, Date, Time, and Duration operations
- Boolean and URI validation
- List operations

3.3.5 Summary

This section has presented the state of art in Knowledge Representation and its importance to characterize and to store the knowledge of a domain in a computer readable format. A fundamental part in KR is the utilization of a language to represent the knowledge of a domain. Description Logics languages have the necessary properties to obtain the required expressiveness when modeling a domain of interest.

Ontologies can provide the required vocabulary and semantics in knowledge representation. If ontologies are well defined using a DL language, their power of expressiveness can cope the lack of sharing understanding, for example, in the QoS management domain. Indeed, ontologies facilitates the communication between people, interoperability among systems while, at the same time, they keep its properties like re-usability, reliability and specification.

This section has also presented the Web Ontology Language (OWL), specifically OWL-DL, which offers a rich expressiveness and reasoning capabilities like subsumption, equivalence, consistency, instantiation checking, and inferences of new conclusions. Certainly, OWL-DL has proven its effectiveness given the fact that it is the standard *de facto* for knowledge representation in the Semantic Web. Even when OWL was oriented to semantic web applications, its use has been spread to different domains. We have chosen the OWL-DL as a DL language to develop our ontology-based models because of its properties as well as its reasoning and rules capabilities.

The following section presents the ontology-based models included in MODA.

3.4 MODA

We think that by using semantic models, we can create a semantic space shared by the different actors in a home network context. As we have seen in the previous section 3.3.2.1, ontologies can be a formal approach to describe the knowledge of a domain and, also, to share the semantics between the involved actors. The support of a formal or semi-formal language is important in order to produce formal or semi-formal ontologies. As we have seen in the previous section, OWL-DL is a good choice, given its characteristics, if we want to obtain formal descriptions for knowledge representation.

The goal of this section is to introduced one of the two main contributions of this thesis: an ontology-based approach that we have named MODA. MODA stands for *Multimedia Ontology Driven Architecture*. The main goal of MODA is to allow home users, application programmers, and home-network providers to have a common space, where they can share the semantics about user preferences, multimedia services, and QoS parameters and requirements of multimedia applications.

Before starting the creation of MODA, we think that an important question to answer is *what are the knowledge requirements that should drive the creation of semantic models of MODA?* In other words, *what are exactly the parts of the domain of knowledge we want to model in MODA ?* The following subsection presents the use cases representing the functional requirements for MODA in order to provide QoS in the context of home networks.

3.4.1 Requirements for MODA for QoS Management

The necessity of a semantic common space, where all the actors involved in the QoS management can share their points of view regarding the QoS they want in a home network, leads us to propose a set of use cases in order to define the functionality that MODA should provide.

Figure 3.4 shows the general uses cases of MODA for QoS management in a home network. The goal of figure 3.4 is to give a description of the functionality we believe MODA should provide to the involved actors. In the following paragraphs each one of the use cases will be explained.

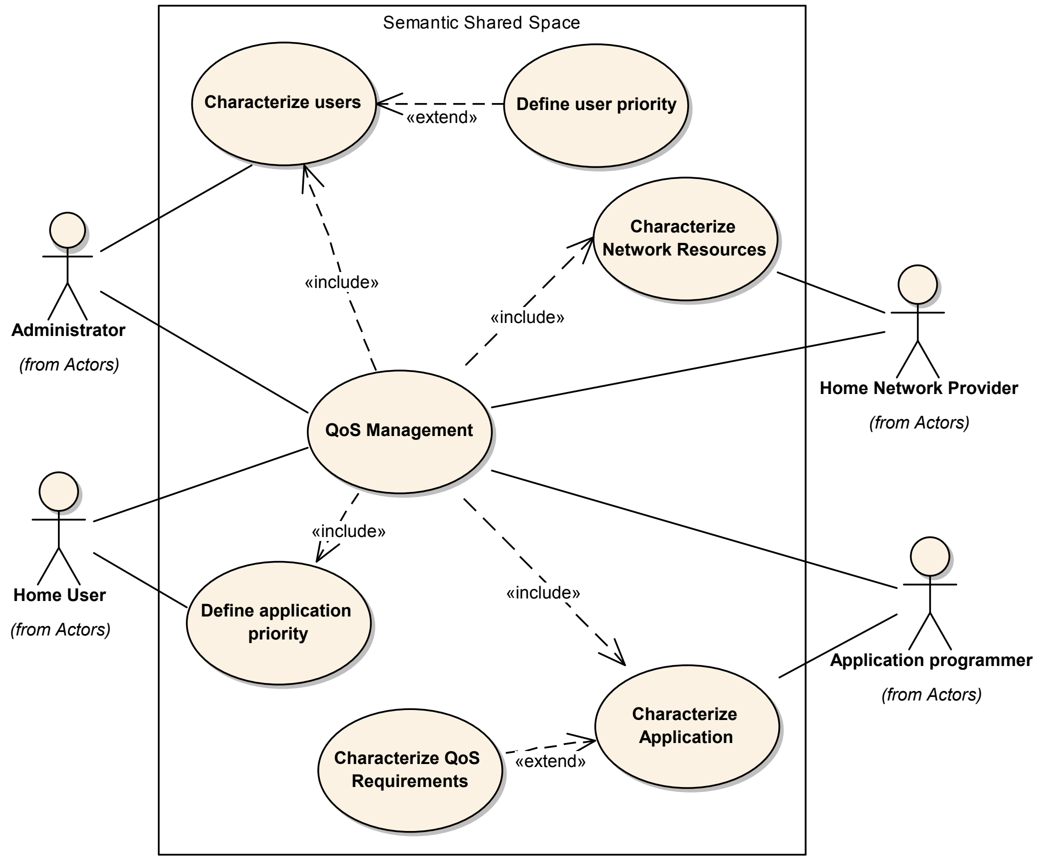

Figure 3.4: Use Cases of MODA for QoS Management

1. Use Case Name: Characterize Users.
Actor: Administrator.
Use Case Description: Within a home network, the administrator user and only him/her should be able to characterize other home network users. A user charac-

terization includes data user as its name, but also, the administrator should be able to describe the user priority.

2. Use Case Name: Define user priority.
Actor: Administrator.
Use Case Description: A home network administrator should be able to describe the priority of a user. In a home network, one can find several users and each of them may have a different priority for the assignations of network resources. This information should be understood by a QoS manager when assigning the network resources.

3. Use Case Name: Characterize Network Resources.
Actor: Home Network Provider.
Use Case Description: The home network service provider should be able to describe the home network resources such as network capacity, IP addressing of the devices, IP addressing type, etc.

4. Use Case Name: Characterize Applications.
Actor: Application Programmer.
Use Case Description: The application programmer should be able to describe multimedia applications and their requirements. To this end, the application programmer should be able to indicate, for example, the number of entities participating in the communication (point-to-point), the direction in which the information is sent (pushed from producer to consumer or pulled by the consumer from the producer), multimedia transport protocols, transport communication ports, etc.

5. Use Case Name: Characterize QoS Requirements.
Actor: Application Programmer.
Use Case Description: In this use case, the application programmer should be able to characterize the QoS requirements of his/her multimedia application. For example, he/she should be able to indicate if a multimedia application and the flows of data produced and/or consumed are real-time constrained.

6. Use Case Name: Define application priority.
Actor: Home User.
Use Case Description: The home user should be able to assign a preferred priority. By application priority assigned by its user we mean the preference of the user for its application and not the priority of the type of traffic generated by the application (as meant in standards like IEEE 802.1D, IEEE 802.1Q, and IEEE 802.11e).

7. Use Case Name: QoS Management.

Actors: Home User, Administrator, Home Network Provider, Application Programmer.
Use Case Description: This is the main use case. This use case includes all the other use cases presented before. The characterization of applications, users, and network resources from the different actors' points of view should allow the alignment of all the concepts used in the domain of knowledge. Particularly, the alignment of concepts by its meaning may allow a QoS Management driven by user preferences, application constrains and network capabilities.

The above presented use cases for MODA lead us to ask ourselves another important question: *what would be the knowledge source used to construct the needed semantic models or ontologies to be shared by these actors?* Evidently, we want to characterize, specifically the knowledge in the domain of QoS management in the context of home networks. But, *should we start by building ontologies from our own understanding ?* We think not. We want to take advantage of the already existing efforts and knowledge in the domain of QoS management.

Currently, international organizations like IEEE, ISO and ITU have produced an important set of international standards and recommendations, some of them covering subjects in the domain that we are interested in. Thus, we have made a search between standards, recommendations, and RFCs that are focused in our domain of interest and, at the same time, that are in line with the presented use cases. We have decided to construct the semantic models of MODA based on such standards in order to take advantage of its world-wide acceptation and standardization.

3.4.2 Ontology-based models of MODA

In this section the different models included in MODA are presented. MODA is an ontology driven approach. As a matter of fact, MODA follows the methodology proposed by the Model Driven Architecture [OMG03] and it is combined with the Ontology Driven Architecture [TPO^{+}06].

Figure 3.5 helps to better understand the MDA-ODA approach that we are using for MODA. In this figure, we can see the three axes that compose MODA. The first axis includes the MDA approach, thus here we find the different levels of abstraction CIM, PIM, PSM proposed by MDA. The second axis considers the ODA approach, thus it includes the models of each domain of knowledge of our interest. As consequence, at CIM level, we find the world-wide accepted standards and specifications that are the basis of the ontology-based models produced at PIM level; at PSM level, we find the ontology instances of PIM's level as well as the models of the platforms like UPnP QoS. Finally, the third axis includes the actors' perspective for which the models are selected, created, and instantiated.

For easier reading, the ontology-based models of MODA are presented as follows: first, we explain the standards on which the ontology is based (CIM level). Second, we present the corresponding ontology-based model (PIM level). The platform specific models including the instances of ontology-based PIMs are presented in chapter 5 as part of the deployment and evaluation of MODA.

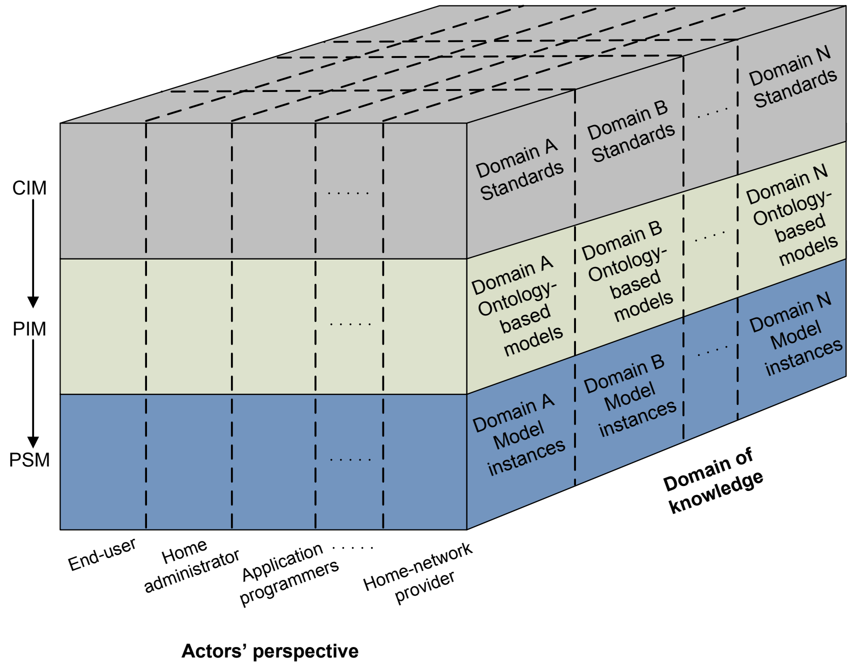

Figure 3.5: Combined MDA and ODA for MODA

3.4.3 ITU-T Recommendation F.700

ITU-T Rec. F.700 [Uni00] proposes a framework for characterizing multimedia services. From a functional point of view this standard provides a methodology for the development of multimedia services considering needs of both final users and service providers. The approach proposed in ITU-T Rec. F.700 is based on a four-level model, in a top-down order: Application, Service, Communication Task, and Media Component levels.

The Application level describes the functional characteristics from a user point of view. The service level includes services or tools that satisfy the functional requirements of the application level. Services like QoS, security, or intercommunication are defined in the service level. According to the model, the construction of services is done by combining communication tasks and coordinating their interactions. Downwards, at the communication task level, communication tasks are defined as functional entities of multimedia services, and also they handle media components in order to transport information. Functions like transfer, storage, and switching are defined in this level.

Finally, at the bottom of the model, the media component level deals with the multimedia aspects of the services by describing the monomedia components such as audio, video, etc, of user information. At this level, functions like capture, coding, presentation, etc. are established. Regarding control activities, ITU-T Rec. F.700 proposes a control and processing plane, which interacts with the service, communication tasks, and media component levels through middleware service elements. Figure 3.6 depicts the model proposed by ITU-T Rec. F.700.

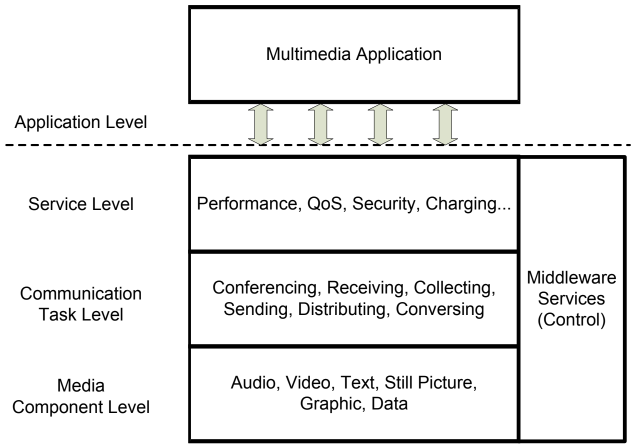

Figure 3.6: ITU-T Rec. F.700 Multimedia services reference model.

In a top-down approach, ITU-T Rec. F.700 suggests the decomposition of a multimedia service into communication tasks controlled by user and/or service providers. In a bottom-top approach a communication task can be viewed as the means to gather the media components required for multimedia services. Since communication tasks are the means for composing multimedia services and gathering the media components, their description is particularly important. ITU-T Rec. F.700 proposes three attributes: communication configuration, control entity, and information flow in order to describe generic communication tasks like sending, conversing, conferencing, distributing, collecting, and receiving.

In other words, ITU-T Rec. F.700 gives the possibility to describe and to construct multimedia services in an automatic manner, for instance by developing an ITU-T Rec. F.700 ontology which describes multimedia services composition. This ontology, being computer readable, facilitates automatic configuration and/or construction of the multimedia services. Further, when integrating other standards along with ITU-T Rec. F.700, one can describe/construct multimedia services considering simultaneously aspects like QoS (ITU-T Rec. X.641), and/or user requirements for delay and information loss (ITU-T Rec. G.1010).

3.4.3.1 ITU-T Recommendation F.700 Ontology-based Model

In this section we describe the ITU-T Rec. F.700 ontology. An important part of the ITU-T Rec. F.700 is the description and construction of multimedia services based on their communication tasks. The ITU-T Rec. F.700 ontology is focused on the description of communication tasks through their attributes. In this manner, a communication task has:

- Communication configuration. This class in the ontology is used to express if the communication is point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint.
- Symmetry of information flow (FlowSymmetry). It allows specifying the direction in which the information is sent.
- Transmission Control Entity (TransControlEntity). It allows saying who is controlling the transmission of the information (e.g. source and/or sink).
- Communication delay. This concept characterizes the type of delay supported by the communication task e.g., real time.
- Media. This concept characterizes the media (multi or mono), mandatory or optional ones, transmitted by the communication task as well as the quality level of the media.
- Time continuity. This class allows expressing if the communication task is buffer capable or not.
- Media interrelation. It allows to characterize the relations between media. It specifies if there is some synchronization between media (e.g. lips or subtitles synchronization), or symmetry in order to indicate bidirectionality of the same media type, or conversion between media in order to indicate when a media is converted into another type of media, i.e. when graphics are converted into still pictures.

In Figure 3.7 we can observe the relations between the main classes of the proposed ITU-T F.700 ontology. Using the before defined elements, we can model a `Communication Task` (accordingly to ITU-T F.700) as follows:

$$\begin{gathered}\text{CommunicationTask} \equiv \top \sqcap \\ (\exists\ \text{hasConfiguration.CommunicationConfiguration}) \sqcap \\ (\exists\ \text{hasCommDelay.CommunicationDelay}) \sqcap \\ (\exists\ \text{hasFlowSymmetry.FlowSymmetry}) \sqcap \\ (\exists\ \text{hasMedia.Media}) \sqcap \\ (\exists\ \text{hasTimeContinuity.TimeContinuity}) \sqcap \\ (\exists\ \text{hasControl.TransControlEntity})\end{gathered}$$

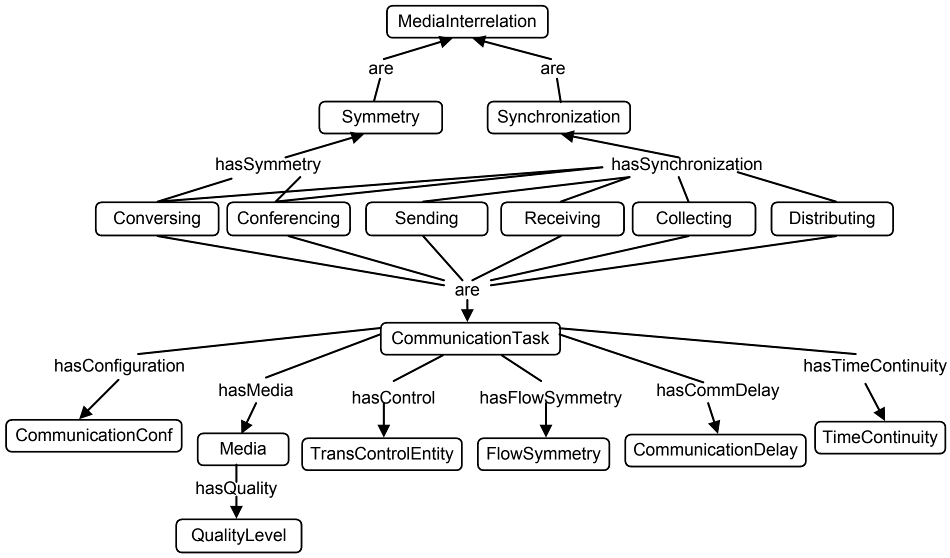

Figure 3.7: ITU-T F.700 Ontology-based Model.

3.4.4 ITU-T Recommendation X.641

ITU-T Recommendation X.641 [Uni97] provides concepts, terminology and definitions relating to QoS in order to supply a common approach and vocabulary for QoS in an Information Technology (IT) environment. In order to do so, the ITU-T Rec. X.641 framework introduces the concept of QoS characteristics; and it defines several QoS characteristics and their relationships. In addition, this framework describes how QoS requirements can be expressed as well as the QoS mechanisms used as components of QoS management functions, that help to achieve QoS requirements. The following paragraphs explain in detail the main concepts of the recommendation.

QoS Characteristic. ITU-T Rec. X.641 defines a QoS characteristic “as a quantifiable aspect of QoS, which is defined independently of the means by which it is represented or controlled”. QoS characteristics are intended to be used to model the actual, rather than the observed, behavior of the systems that they characterize.

The recommendation adds three concepts about QoS characteristics: generics, specialization, and derived QoS characteristic. A QoS characteristic can be a generic QoS characteristic when it applies to a variety of circumstances. For a generic characteristic, one or several specializations can be defined in order to make the characteristic useable in practice. Finally, a derived characteristic is defined as a function(s) of others QoS characteristics e.g. statistical derivations like variance or mean.

In order to define new QoS characteristics, ITU-T Rec. X.641 proposes a descriptive technique which states that the definition of a QoS characteristic includes:

- a name for the characteristic;
- a definition explaining its purpose;
- a statement of how the characteristic is quantified, providing the units in which the values are expressed;
- statistical Derivation (if any)
- Specialization (if any)
- further information (optional)

QoS requirements and QoS parameters. User requirements are quantified and expressed as set of QoS requirements. When conveyed between entities, QoS requirements can be expressed as QoS parameters. However, the QoS requirements are expressed in QoS context when they are retained in an entity. The recommendation defines a QoS requirement as the QoS information that expresses a part or all of a requirement to manage one or more QoS characteristics.

QoS management functions are triggered when QoS requirements are not achieved. It is through the levels or values of QoS parameters that a QoS management function is performed when a QoS requirement is not accomplished. In other words, the values or levels of QoS parameters play an important role. ITU-T Rec. X.641 proposes semantics for the QoS parameters when agreeing QoS requirements. In order to understand the meaning of the values of QoS parameters the following limits, thresholds, and target are proposed (see Figure 3.8).

- Controlled Highest Quality (CHQ) limit. The value of a QoS parameter that overcomes this limit means that there is an over provisioning which it is not necessary for the good performance of the system.
- Upper Threshold. It is an identified point at which specific actions are defined. Evidently the upper threshold (High quality) value must represent QoS greater than or equal to the lower threshold (Lower quality). When the upper threshold is crossed in the increasing direction, a set of actions associated to the threshold may carry out. The upper threshold could be single or doubled valued. In the case of single valued threshold, it is necessary to indicate the meaning of the value in order to understand it whether as “high quality” or “low quality”, as well as to determine the direction of crossing the threshold.

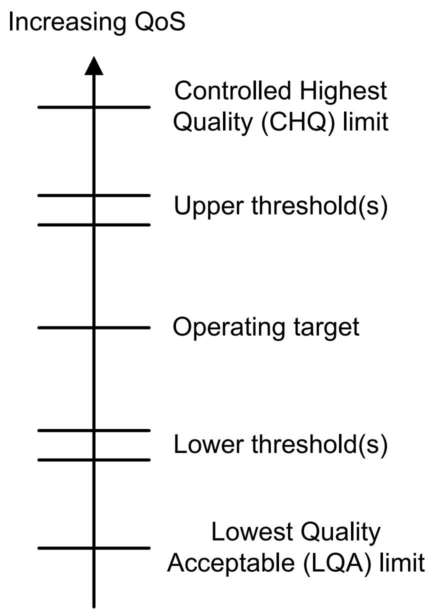

Figure 3.8: Limits, thresholds and target parameters of QoS requirements.

- Operating Target. It is defined as a negotiated or imposed level at or near which QoS is agreed to be maintained.

- Lower Threshold. The idea of using the lower threshold is similar to the upper threshold but in the other direction. In this sense, it is possible to have a set of actions associated to the threshold when it is crossed in the decreasing direction.

- Lowest Quality Acceptable (LQA) limit. The value of a QoS parameter under this limit means that the QoS requirement is no longer accomplished; in other words, the QoS parameter should not fall below this limit. The QoS management functions triggered when crossing this limit depend on the agreement level entered for this requirement.

Levels of Agreement.The actions that service providers and service users agree to take in order to maintain agreed levels of QoS may depend on the level of agreement imposed or negotiated. This recommendation suggests three levels of agreement, namely:

- Best effort. It is the weakest agreement. There is no assurance that the agreed QoS will be provided. Also there is no commitment to monitor the QoS achieved or to take any action.

- Compulsory. The service must be aborted if the QoS degrades below the agreed level. The agreed QoS might be intentionally degraded in order to satisfy priority services.

- Guaranteed. The agreed QoS must be guaranteed. It means that the service will not be initiated unless it can be maintained within the specified limits.

3.4.4.1 ITU-T Recommendation X.641 Ontology-based Model

The ITU-T Rec. X.641 ontology allows us to describe the QoS requirements of multimedia services. According to the ITU-T Recommendation X.641, the ontology proposes the following concepts and relationships:

- QoSCharacteristic. This class in the ontology represents a generic QoS concept to be measured in the system. An individual of QoS Characteristic has a name, a description, a measurement unit, and may have a derivation e.g. statistical derivation.

- LevelOfAgreement. This class has 3 instances representing the best-effort, where no actions are taken in order to provide or maintain the agreed QoS. Compulsory, which means that the services must be aborted when QoS requirement is not satisfied or actions are triggered to solve the problem. Finally, guaranteed level where QoS requirement always guaranteed.

- Measurement. It allows to establish the unit measure for the parameter value of the QoS requirement.

- QoSRequirement. This class in the ontology represents a QoS requirement. A QoS Requirement may relate to a number of QoS Characteristics.

- QoSValue. A QoS Requirement may have QoS parameter values, namely: a target, limits and threshold values. We use the class QoSValue in order to model such QoS requirement values

    - Target value. This property represents the target value of a QoS requirement if any.
    - Limit value. Property which allows the lowest quality accepted and the controlled highest quality values to be specified.
    - Threshold value. This property allows to set the values of the upper and lower thresholds for a QoS requirement.

Figure 3.9 depicts ITU-T Rec. X.641 ontology-based model. In the ITU-T Rec. X.641 ontology, we have defined the `QoSRequirement` concept as follows:

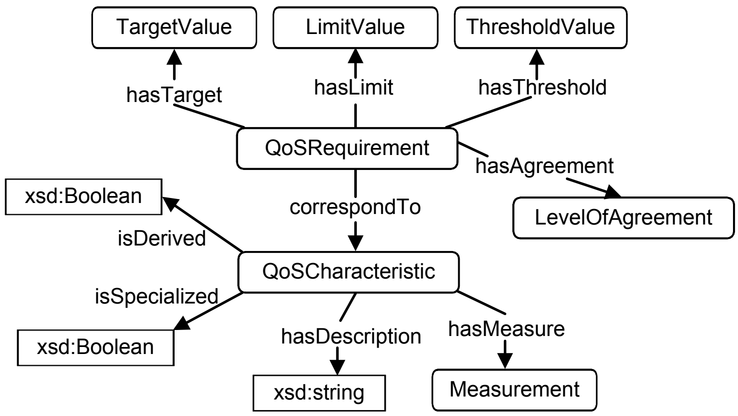

Figure 3.9: ITU-T X.641 QoS Ontology-based Model

$$
\begin{gathered}
\text{QoSRequirement} \equiv \top \sqcap \\
(\exists\ \text{correspondTo.QoSCharacteristic}) \sqcap \\
(\exists\ \text{hasLevelOfAgreement.LevelOfAgreement}) \sqcap \\
(\exists\ \text{hasQoSValue.QoSValue})
\end{gathered}
$$

Where the `QoSValue` concept is described as follows:

$$
\begin{gathered}
\text{QoSValue} \equiv \top \sqcap \\
(\exists\ \text{hasMeasure.Measure}) \sqcap \\
(\exists\ \text{hasOperatingTarget.string}) \sqcap \\
(\exists\ \text{hasControlledHighestQuality.string}) \sqcap \\
(\exists\ \text{hasLowestQualityAcceptable.string}) \sqcap \\
(\exists\ \text{hasLowerThreshold.string}) \sqcap \\
(\exists\ \text{hasUpperThreshold.string})
\end{gathered}
$$

And the `QoSCharacteristic` concept is described:

$$
\begin{gathered}
\text{QoSCharacteristic} \equiv \top \sqcap \\
(\exists\ \text{hasName.string}) \sqcap \\
(\exists\ \text{hasDescription.string}) \sqcap \\
(\exists\ \text{isSpecialized.boolean}) \sqcap \\
(\exists\ \text{isDerived.boolean})
\end{gathered}
$$

3.4.5 Priorities of Users and Applications

In order to deal with priorities of users and applications, we have been inspired by the manner most standards and specifications manage traffic priorities. This subsection presents a recapitulative about the standards and protocols dealing with priorities in MAC and network layers that we used as models to develop our ontology-based model.

At the MAC layer, standards like e.g., IEEE 802.1D or IEEE 802.11e, make a traffic classification based on delay and loss requirements in order to assign priorities (traffic importance) to data flows. Even when user priority is considered in these standards, it usually is optional, e.g., standards like IEEE 802.1Q or IEEE 802.11 in which, originally, it was not even considered. Even when user priority is used, for example in IEEE 802.11e or Wi-Fi Multimedia (WMM), such priority utilization still depends on the upper layers (e.g. network layer) treatment. As a consequence, user priority or traffic classification should be addressed in both network and MAC layers.

We think there is a semantic problem with the term *user priority* in all these layer 2 standards because the meaning of *user priority* is used to map its value to a traffic class e.g. background, video, or audio traffic. This means that the *user priority* value does not actually correspond to the user but to the traffic the user is generating. Indeed, it is not simple to consider the *priority* of a user at this level of the network stack.

At network layer, DiffServ uses the Differentiated Services Code Point (DSCP) field in order to classify packets. DSCP replaces the Type of service (TOS) field. But, whether it is TOS or DSCP, network packets need to be marked with the corresponding priority in order to be classified.

Indeed, in an End-to-End path, it is quite complex to consider truly user priorities at both layer 2 and layer 3 protocols. However, in the context of a Home Network, not only the consideration of user priority is possible but also, user priority for QoS management gains importance given the fact that in a home network there are several type of users like for example administrator user, visiting user, etc., or network utilization profiles, like administrator user *working* or visiting user *playing*.

The specification of UPnP QoS v3 considers the assignation of priorities to network users and traffic by using an integer number representing the user importance. User priority is considered when there are no sufficient network resources, allowing new traffic flows to be prioritized or admitted over other flows belonging to users with lower priorities. Traffic priority is used to map the importance of the flow at layer 2 in the UPnP QoS Device.

3.4.5.1 Users and Application's Priority Ontology (UAPO)

We have created a User and Application Ontology-based model in order to express the priorities that users have themselves and the priority they assign to their applications. This ontology characterizes the semantics of user and application priority, and also it is used to describe the priority preferences between home users and between applications. The concepts defined in UAPO are the following:

- User. This class permits the description of the user. There are two types of user: human (Human User) and non human (Non Human User) users. A human user can have a name, username, and e-mail as properties. Also, a user can have a priority which represents the user importance in the system. In consequence, a user e.g., home network administrator can express priorities between other users.

- User Role. This class represents the different roles a user can play, for instance: administrator or home user.
- SDO:Application. This class belongs to the session description ontology. Here the property *hasPriority* is defined in order to allow the users to define a priority for their applications. Thus, the user can express the maximum and minimum numerical priority values for his/her applications as well as the limits of the priority values (max and min). Finally, an instance of `SDO:Application` is a ITU-T F.700 `ComunicationTask`.

We have described a user as follows:

$$\begin{gathered}\text{User} \equiv \top\sqcap \\ (\exists\ \text{hasUserRole.UserRole}) \sqcap \\ (\forall\ \text{hasUserRole.userRole}) \sqcap \\ (= 1\ \text{hasName.string}) \sqcap \\ (= 1\ \text{hasLogin.string}) \sqcap \\ (= 1\ \text{hasPassword.string}) \sqcap \\ (= 1\ \text{isSpecialized.boolean}) \sqcap \\ (= 1\ \text{isDerived.boolean})\end{gathered}$$

Figure 3.10 shows the concepts and their relationships in UAPO.

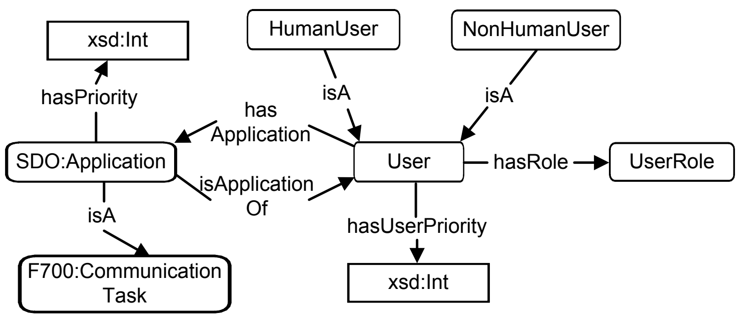

Figure 3.10: User and Application Ontology.

At this point, we have defined the ontologies which allows us to model:

- A distributed multimedia application,

- The QoS requirements; and
- The User and Application's Priorities.

The following subsection presents the ontology which allow us to characterize hosts, networks and its resources.

3.4.6 Device Description

There have been several efforts to develop ontologies for device description. An early approach to device description was proposed by the Foundation for Intelligent Physical Agents (FIPA)[2]. FIPA has developed the Device Ontology Specification, which is a frame-based ontology that characterize device capabilities.

The Fieldbus foundation, PROFIBUS Nutzerorganisation, and HART Communication Foundations has developed the Electronic Device Description Language (EDDL)[Com10] [3]. EDDL is a language for describing the properties of automation system components. EDDL is used for describing the operation and parametrization of devices mainly produced by industry automation using text files in ASCII format. Still in automation systems domain, Dibowski and Kabitzsch [DK10] propose Ontology-based Device Descriptions (ODDs) as an approach to device descriptions using OWL. All of these works are aimed at being used for the automation of systems. Thus, they take into account a lot of devices characteristics and parameters that are not used in for example a printer or TV display in a home network.

Focused on the WWW, the W3C consortium has proposed as a Working Draft the Composite Capability/Preference Profiles (CC/PP) 2.0[4] aimed at defining profiles based on RDF. According to CC/PP, A CC/PP profile "is a description of device capabilities and user preferences that can be used to guide the adaptation of contents presented to that device". In other words, CC/PP is aimed at describing device's delivery context and use information to guide the adaptation of contents presented to that device.

In the following subsection, we present our ontology aimed at characterizing the different devices located in a home network. We have taken into account the different point of views of the different works presented in this section. However, such works consider a lot of device characteristics that are out of our scope. Thus, we have decided to develop an Host-Network ontology in order to describe the devices one can find in home network. We have also taken into account the information needed by protocols like SDP and SIP.

[2] http://www.fipa.org/specs/fipa00091/PC00091A.html
[3] http://www.eddl.org
[4] http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/

3.4.6.1 Host-Network Ontology (HNO)

We have defined the Host-Network ontology as a simple but complete representation of network devices in a home network. HNO defines the following concepts, properties and relationships:

- Host. The host concept characterizes every node in the network (home network). First, a *Host* has a host name which is its identifier in the network and it is unique in the network (functional property). A host may have a *gateway* which is also a host in the network. A host is located within a networks as the property *isLocatedIn* represents it. A host has also a network address (*NetworkAddress*).

- Network. A network contains *hosts*. It has a *networkID* given by its IP network address. Also, it has a *NetworkType* which can be, for example, "IN" meaning "Internet" (public) or it can be "Private".

- Network Address. A *NetworkAddress* belongs to a only one *Host* (functional property *isNetworkAddressOf*). A network address have a type represented by *hasType* which can be "IP4" or "IP6". A network address has also an addressing type indicated by the property *hasAddressingType*. Its values can be "Unicast", "Multicast", "Broadcast", and "Anycast".

  Figure 3.11 depicts the Host-Network ontology.

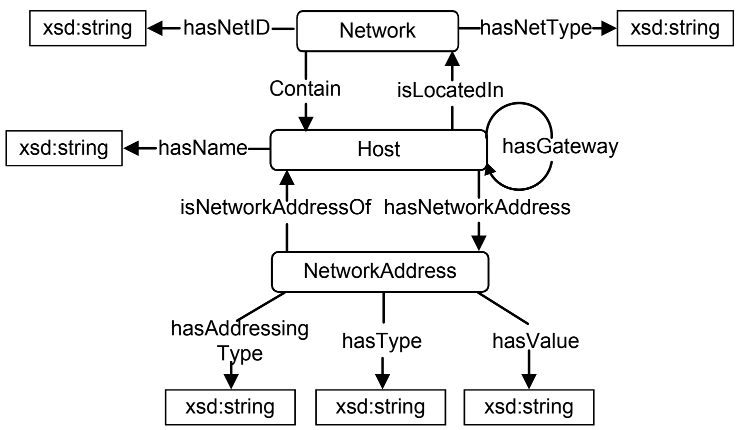

Figure 3.11: Host-Network Ontology.

Now we want to put together all their instances during the session establishment of the distributed multimedia system. In order to do so, we have developed a model based on the session description protocol (SDP) [HJP06], and we have integrated it to the previous ontology-based models. The following subsection presents in detail the session description model we have developed.

3.4.7 MMUSIC Session Description Protocol (SDP) and SDPng

The Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task Force (IETF) has been responsible for the specification of the more widely used session protocols for multimedia systems. The MMUSIC Working Group has proposed the Session Description Protocol [HJP06]. The Session Description Protocol (SDP) provides a common representation for expressing media and session descriptions. SDP proposes an entirely textual data format to maximize portability among distributed applications. SDP is intended to describe multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. For this purpose, SDP has to carry information like media details, transport addresses, and session description metadata to participants of the session (e.g. streaming, videoconference, VoIP sessions). Indeed, the idea of using SDP is to describe the session information in a standard format. Currently, SDP information is utilized in the following protocols:

- With the Session Initiation Protocol (SIP) [HSSR99] for creating, modifying, and terminating sessions particularly in VoIP.
- With the Real Time Streaming Protocol (RTSP) [SRL98] in order to control on-demand delivery of data with real-time characteristics.
- With the Session Announcement Protocol (SAP) [HPW00] for distributing session description information to potential participants in multicast sessions.

The diversity of use of SDP has led to requirements for which SDP was not originally designed. In order to fill these gaps, several extensions to SDP have been proposed e.g., the offer/answer model with the SDP [RS02] and grouping of media lines in the SDP [CS10].

The MMUSIC Working Group has proposed SDPng as the SDP successor. SDPng considers besides the session description information, the dynamic aspects (e.g. parameters and configuration) of interactive sessions. One important technical characteristic of SDPng is the use of an XML-based syntax in order to increase the expressiveness required for achieving dynamic aspect like session negotiation.

Taking SDP and SDPng as the basis for developing a session description model based on ontologies enables not only to describe the session information (formatting aspect) but also allows to integrate the meaning of information (semantic aspect). Semantics can be used to achieve a better dynamic session configuration.

3.4.7.1 SDO: Session Description Ontology

In this subsection, a session description ontology (SDO) based on the SDP and SDPng is presented. The SDO makes use of the already presented ITU-T Recommendation F.700 ontology. Figure 3.12 illustrates the concepts and relationships of the SDO. The following paragraphs describe in detail the purpose of the classes of

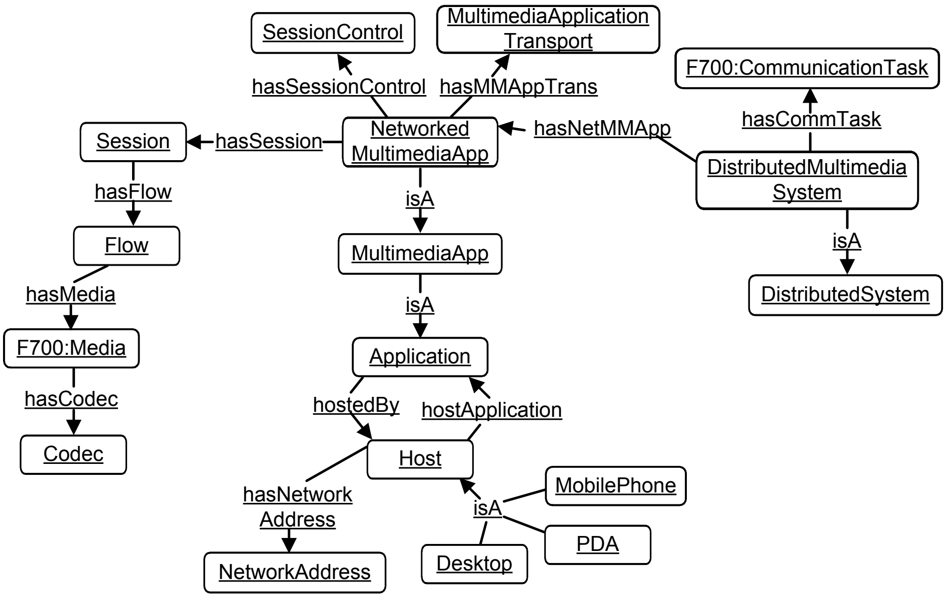

Figure 3.12: Session Description Ontology .

the session description ontology:

- Application. This class characterizes the applications. Application class is used by the user/application priorities ontology.

- Multimedia Application. This class characterizes multimedia applications which are subclass of the Application class.

- Networked Multimedia Application. Given the fact that multimedia applications may be local or networked, this class characterizes networked multimedia applications (clients, servers or peers).

- Distributed Multimedia System. This class characterizes a distributed multimedia system that is composed by two or more networked multimedia applications.

- Host. This class describes the device that is hosting the application. It includes a number of subclasses representing several devices like mobile phones, PDAs, and Desktop. Also, information about the characteristics of the device is considered (e.g. display resolution).

- Session. This class describes some components of a networked multimedia session. For example, using the session class, the media flows of the networked multimedia applications are described.

- Session Control. Individuals of this class are session control protocols like RTSP, SIP or SAP. Certainly, the session control class allows representing the control session protocol.

- Multimedia Application Transport. This class describes the transport protocols used to transfer the media data. For most of the networked multimedia applications, the RTP/RTCP will be used to accomplish the multimedia data transportation. The MODA framework allows integrating information required by multimedia application transport protocols like RTSP through its corresponding ontology. It is important to remark that the information required to characterize the transport protocol (e.g. RTSP) is available in this ontology e.g. network address, flow information, etc.

- Flow. This class characterizes the flow created by the multimedia application during the session.

- F700:Media. This class belongs to the ITU-T F.700 ontology.

- Codec. This class describes the codec and its parameters used to transmitting the multimedia data. Clearly the codec instance depends on the media in the flow.

3.4.8 Summary

In this section the ontologies included in MODA have been presented. The use cases that model the functional requirements of a semantic common space have been introduced at the beginning of this section. These use cases are presented from each actor's point of view, which help us to delimitate the specific areas of knowledge that should be characterized. The choice of models included in MODA is driven by the non-functional requirements presented in the use cases.

ITU-T Recommendation F.700 allows us to characterize multimedia services. An instance of the ontology of ITU-T Recommendation F.700 can model a distributed multimedia system by using communications tasks such as sending, receiving, conferencing, etc., and defining their corresponding properties like, for example, communication configuration.

Even when QoS requirements are non functional requirements, ITU-T Recommendation X.641 allows us to characterize QoS requirements of multimedia applications from a application programmer's point of view. The ontology of ITU-T

Recommendation X.641 allows defining the expected values of QoS parameters for a given multimedia application.

In order to express the priorities of users and applications, the user and application's priority ontology (UAPO) has been defined. From a user's point of view and, at the same time, taking into account Medium Access Control mechanisms to manage priorities of applications, UAPO allows users to define the priority for each one of their multimedia applications. Also, users having a role of home-administrator can establish priorities between users.

We have also presented an ontology to model network devices and networks. To this end, we have taken several works in the device description domain as references, but we have also based our Home-Network Ontology (HNO) on the information required by session description protocols like SDP and SDPng.

An ontology-based session description has been presented at the final part of this section. We have taken SDP and SDPng as models to build a session description ontology (SDO) that is capable to characterize any existent multimedia session. The goal of SDO is to have a representation at anytime of all multimedia sessions in a home network. This information can be used by management entities, for example, to manage the QoS according to user preferences and to available network resources assigned to existent established multimedia sessions represented by SDO.

3.5 Chapter Conclusion

This chapter is composed of three main sections. The first section presents the state of the art in Model Driven Architectures. Research works like Model Driven Architecture (MDA), Ontology Driven Architecture (ODA), and Ontology Definition Metamodel (ODM) encourages and enforces the utilization of models or ontologies to drive the development and deployment of software systems. Indeed, MDA provides a methodology to produce manual or automated development and deployment of software systems.

The second section presents the state of art in Knowledge Representation (KR), particularly KR with ontologies. Ontologies can provide the required vocabulary and semantics in knowledge representation when they are built with a Description Logic languages. The Web Ontology Language (OWL) is one of the most popular languages used to develop ontologies because of its rich expressiveness and support to reasoning tasks like subsumption, equivalence, consistency, and instantiation checking.

The third section of this Chapter has presented one of the two main contributions of the thesis. We have build a semantic common space we have named MODA. MODA allows the semantic coherence between the points of view of four actors: home user, administrator, application programmer, and network operators. In the context of home networks, MODA is aimed at allowing these four actors to share the same meaning of non functional requirements for QoS provisioning.

Now, by using MODA, we are able to create semantic maps of multimedia sessions in home networks. These maps contain all the necessary information to allow, for example, a home administrator assigning network resources driven by user QoS preferences or by applications QoS requirements. Even it is possible for an administrator user to manually manage the network resources based on the information provided by MODA, we are aiming for the automation of the QoS management.

We think that autonomic computing can provide well adapted solutions to self-manage QoS by enforcing a decision model that uses information provided by MODA. In the next chapter, we present an autonomous framework to manage the QoS as a solution to the problem of QoS provisioning driven by user preference in the context of home networks.

CHAPTER 4

Autonomic Qos Management in Home Networks

Contents

4.1 Introduction

CHapter 3 has introduced the semantic common space proposed by our Multimedia Ontology Driven Architecture (MODA). By using MODA we are able to share knowledge and its semantics, within a home network, between the following actors: home user, home administrator, application programmer, and network operator. It also has presented the different standards that are the basis of the semantic models of MODA. Certainly, the integration of all the semantic models allows us to generate a semantic representation of the various media streams participating in a multimedia session, including its non-functional characteristics.

In home network scenarios, it is particularly necessary to develop QoS frameworks that are able to take into account user preferences e.g. expressing priorities of users and/or applications through friendly user interfaces when assigning shared network resources. Also, such frameworks should be able to keep in repositories all the semantic information in order to be easily used for management purposes. Moreover, this information can help to build QoS policies whether they are driven by service users, service providers, or both of them. Consequently, semantics can be used within decision models that search for efficient or optimal solutions, for example, to make a feasible or an optimal assignation of network resources.

In this chapter, an Autonomic QoS Management Framework in the context of home networks aimed at allowing end users and applications to communicate their requirements and preferences as well as to characterize communication services (network, transport, middleware) is presented. Based on these requirements and preferences, the autonomous provisioning of shared network resources may be performed by well adapted decision models. This chapter is organized in two main sections. The first section presents a state of the art of the Autonomic Computing and Communication paradigms. The second section presents the second main contribution of this thesis: a MODA-based autonomic QoS management framework for home networks.

4.2 Autonomic Computing and Communications

IBM has proposed Autonomic Computing (AC) in order to deal with the growing complexity of management tasks in Information Technology (IT) industry [Hor01]. AC is inspired by the functioning of human body.

As defined by IBM[1], Autonomic Computing is "an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement". In the same manner that the complexity of autonomous nervous system is "embedded" in our body, AC proposes embedding the complexity of management tasks in the system itself.

Implementation and development of Autonomic Systems are not trivial. As a matter of fact, several phases need to be applied in order to produce a fully functional Autonomic System [IBM03]. Figure 4.1 summarizes the five levels of autonomic evolution that a computing system usually passes through in order to finally incorporate autonomic capabilities.

In Figure 4.1, one can see that as a computing system evolves towards autonomic behavior, human intervention in the system management reduces considerably. The autonomic paradigm is not only reserved to computing capabilities of systems. Indeed, aspects as networking and communications are also concerned with similar autonomic goals.

[1] http://www.research.ibm.com/autonomic/overview/faqs.html

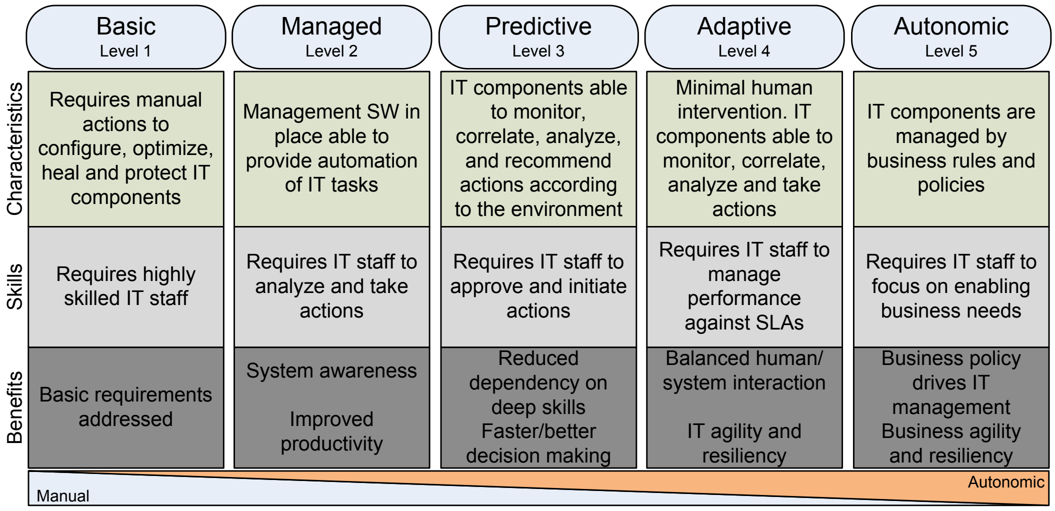

Figure 4.1: The five levels of autonomic evolution. From [IBM03]

Autonomic Communications. Inspired by Autonomic Computing, Autonomic Communications deal with the development of self-managing network and communication infrastructures. Indeed, Autonomic Communications aim at proposing solutions that cope with the complexity of network and communication systems management. As stated in [BZ07], Autonomic Communication focuses on distributed systems and management of network resources at both infrastructure level and user level.

Current communications scenarios have properties like ubiquity, mobile connectivity, device and platform diversity. Additionally, complexity factors, like dynamics of the network, decentralization and control, make of system management a complex task to be carried out [QZ08].

Despite of Autonomic Computing and Autonomic Communication share the same principles, they have differences. On the one hand, Autonomic Computing is more oriented to application software and management of computing resources [BZ07] with the objective of reducing cost of ownership of complex IT. On the other hand, Autonomic Communication seeks for a deep rethinking of communication, networking, and distributed computing paradigms to face increasing complexities and dynamics of current network scenarios. Indeed, as stated by Dobson [DDF$^+$06], next-generation networks are expected to grow more chaotically, radically distributed and decentralized, considering actual trends to mobility, roaming, ubiquity, and pervasive. Integrating new technologies will be more difficult from service provider's point of view.

There are four general self-* properties that constitutes the self-management of Autonomic Computing and Autonomic Communication: self-configuration, self-healing, self-optimization, and self-protection. The autonomic paradigm is open and accepts other self-* functionalities or attributes like self-monitoring, self-adapting, self-planing, or self-learning [SPTU05, Ste05, Tia03]. The following subsection details the four general self-management functions proposed by Autonomic Computing.

4.2.1 Self-Management

As stated in [IBM03], in an autonomic environment, system components have self-managing functions such as self-configuration, self-healing, self-optimization, and self-protection. These self-management operations allow autonomic systems to maintain and adjust their operations in the presence of both expected an unexpected changes in the environment [KC03]. IBM defines four general self-managing activities as follows:

- Self-configuration. It is a self-management task that allows self-configuring components adapting dynamically and automatically to changes in the system context. Self-configuring components are driven by policies provided by IT professionals. Policies express high level objectives that indicate what the user wants, but not how such objectives should be accomplished.

- Self-healing. Autonomic elements that implement this self-property are able to detect, diagnose and react to malfunctions. Self-healing allows the autonomous execution of policy-based actions in order to repair localized software and hardware problems.

- Self-optimization. This property allows autonomic systems to constantly look for the improvement of their performance. Self-optimization components constantly monitor system parameters and adjust them automatically in order to allow the system to be more efficient in performance, resource utilization, etc.

- Self-protection. Self-protection components are able to anticipate and detect problems based on their knowledge (reports, logs, etc); but also, they are capable to identify and protect against hostile behaviors, then they can take actions to enforce security and privacy policies.

The above mentioned self-* functionalities are achieved by Autonomic Elements (AE) defined in an Autonomic Computing Architecture. AE's are the modular components in Autonomic Systems. The following subsection describes the components that compose the Autonomic Computing Architecture as well as its functionalities.

4.2.2 Autonomic Computing Architecture

An Autonomic System is composed of *autonomic elements* (AE). AEs collaborate in order to provide one, several, or all of the self-* functionalities in the Autonomic System. In turn, an AE is composed of an Autonomic Manager (AM) and one or more Managed Elements (MEs). Through the AM, AEs manage their internal behavior as well as their interactions with other AEs according to the policies defined by humans or other AEs. Policies and/or rules are stored in the knowledge source.

Figure 4.2 presents a component diagram illustrating the AC architecture. In this figure, we can see that Autonomic Managers uses sensor and effector interfaces to self-manage the Managed Element.

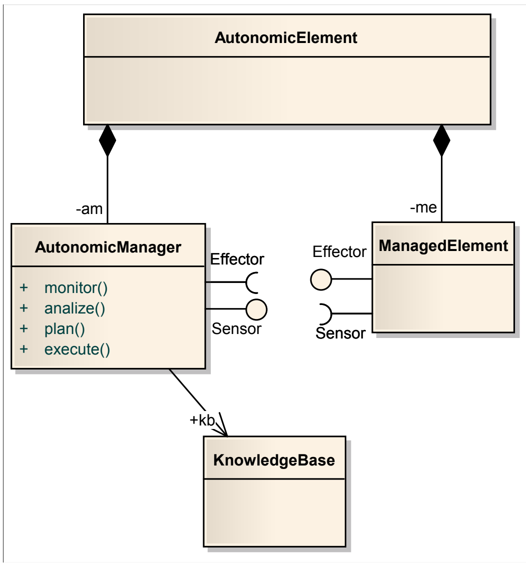

Figure 4.2: Autonomic Computing Architecture

Figure 4.3 shows the functional composition of an Autonomic Element. A Managed Element is controlled through its *sensors* and *effectors* by its Autonomic Manager. ME's sensors and effectors are interfaces that allow an AM to operate on MEs as well as to know its state. The main function of sensors and effectors are:

- Sensor. It is an interface that provides the mechanisms to collect information about the state of an ME by using "get" operations or capturing asynchronous events.
- Effector. It provides the mechanisms that operates on the ME, in other words, the interface that changes the configuration of an element.

Direct or indirect cooperation of autonomic managers is fundamental in an autonomic system because, usually, every AM has a *decision-making* context or

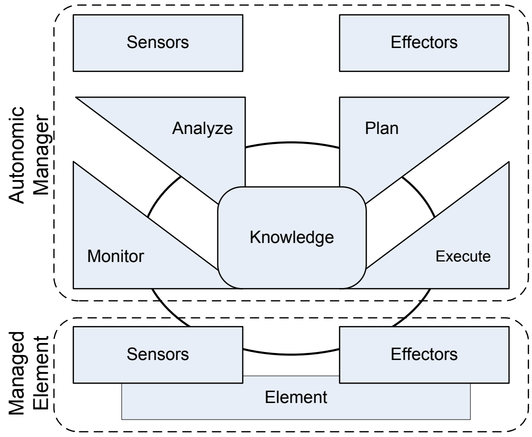

Figure 4.3: Autonomic Element Architecture

management domain. For example, in an autonomic system, it might be necessary to have the cooperation of two or more autonomic managers in order to achieve the self-configuration and/or self-optimization of the system. The *decision-making* process of an AM is driven by its source of knowledge and four functional phases linked in a control loop.

Functional composition of an autonomic manager. The autonomic architecture establishes that autonomic managers must implement a control loop in four functionals phases:

- Monitor: it provides the mechanisms that collect, filter, manage and report details collected from an element.

- Analyze: it provides the mechanisms that correlate data or information and allow the autonomic manager to learn about its context and predict future situations.

- Plan: it provides the mechanisms to structure the actions needed to achieve goals and objectives based on policies.

- Execute: it provides the mechanisms that control the execution of a plan.

All of these functional phases collaborate as needed using asynchronous and synchronous communication techniques, meaning that there is no control flow in a particular order. Also, the four phases share a knowledge source from where they obtain the necessary information to perform their activities.

Knowledge Source. The knowledge source is a repository of information that is created by the user or a third party on behalf of the user in order to define high level policies which will dictate the behavior of the autonomic system. Also, the knowledge used by an autonomic manager could be collected by its monitor function through sensors, for example by registering the notifications it receives from the managed element. The knowledge is linked to the use of policies. As stated in [WHW$^+$04], policies are representations, in a standard external form, of desired behaviors or constrains on behavior.

Knowledge helps autonomic managers to know what to do in a given circumstance. To this end, policies are used as a means of knowledge representation. Based on the work done by Russell and Norving [RNC$^+$09] about intelligent agents, Kephart and Walsh [KW04] propose three policy types for autonomic computing:

- Action policies. They dictate the action that should be taken when the system is in a given current state. They have usually the form IF(condition) THEN(action). Usually, human participation is mandatory to define the policies (IF-THEN clauses).

- Goal policies. Goal policies express a *desire* state rather than an action to be taken. Also, goal policies can specify one or more criteria that describe a set of desired states. The system must compute the action or actions to be executed in order to bring the system to the desired state.

- Utility Function policies. They generalize Goal policies. Utility Function policies are objective functions that express the value of each possible state. The *most desired* state is computed by selecting from the set of feasible states the one that has the highest utility.

The use of policies or rules contributes to the representation of the semantics of a domain of knowledge. Certainly, action policies could help autonomic managers to decide what are the actions to execute in a managed element given the current condition of the system. For example, if the current value of a QoS parameter is below the minimum accepted then a re-assignation of network resource should be made. In other words, a semantic framework like MODA can help to create knowledge sources with the necessary elements to define QoS management policies that would include home users, application programmers and network provider perspectives.

4.2.3 Summary

This section has presented the Autonomic Computing and Communications paradigms. Both paradigms share the same objective: to hide the complexity of management tasks from humans by embedding such complexity in the computing system itself. Particularly, in the case of Autonomic Communication, it is also considered a rethinking of current communication paradigms. We think that end-user participation in QoS provisioning from an autonomic point of view is a contribution to the domain.

Autonomic systems exhibits four general self-properties: self-configuring, self-healing, self-optimization, and self-protection. To this end, autonomic systems have an architecture conformed by autonomic elements that interact in order to achieve the goals defined by the user (IT manager). Autonomic elements include an autonomic manager and managed elements. Autonomic managers perform a control loop divided in four functional phases: monitor, analyze, plan, and execute (MAPE).

The execution of MAPE allows an autonomic manager to operate on managed resources (managed elements) according to high-level policies defined by the administrator. For this purpose, an autonomic manager uses managed element API's composed by sensor and effector interfaces.

As we can see, autonomic paradigm allows for self-management of computing and communication systems. In home network contexts, the development of an autonomic framework would provide the required QoS management capabilities by taking into account QoS preferences from an end-user perspective, multimedia application's QoS requirements, and at the same time, QoS characteristics of the resources offered by the network provider.

The following section describes from a designer point of view the autonomic QoS management based on MODA.

4.3 Autonomic QoS management based on MODA in home networks

In this section, we present the second main contribution of the thesis, which is the Autonomic QoS Management based on MODA for home networks. We have decided to follow the Autonomic Computing approach for the QoS provisioning requirements presented in this thesis. We have seen that the properties proposed by the Autonomic Computing are well suited to solve our problem of QoS management for multimedia applications in the context of home networks. Indeed, by using the Autonomic Computing approach, we are able to limit end-user participation in the QoS management to the definition of high-level directives of QoS preferences.

Figure 4.4 illustrates our QoS provisioning framework based on MODA [EGM10]. This QoS provisioning framework includes a decision model of QoS provisioning that has been defined as an optimal model based on actor preferences and the available services. This model is also able to evolve taking into account changes in the context, e.g., new applications and users, and also to monitor the overall performance of the system in order to take new decisions and enforce them.

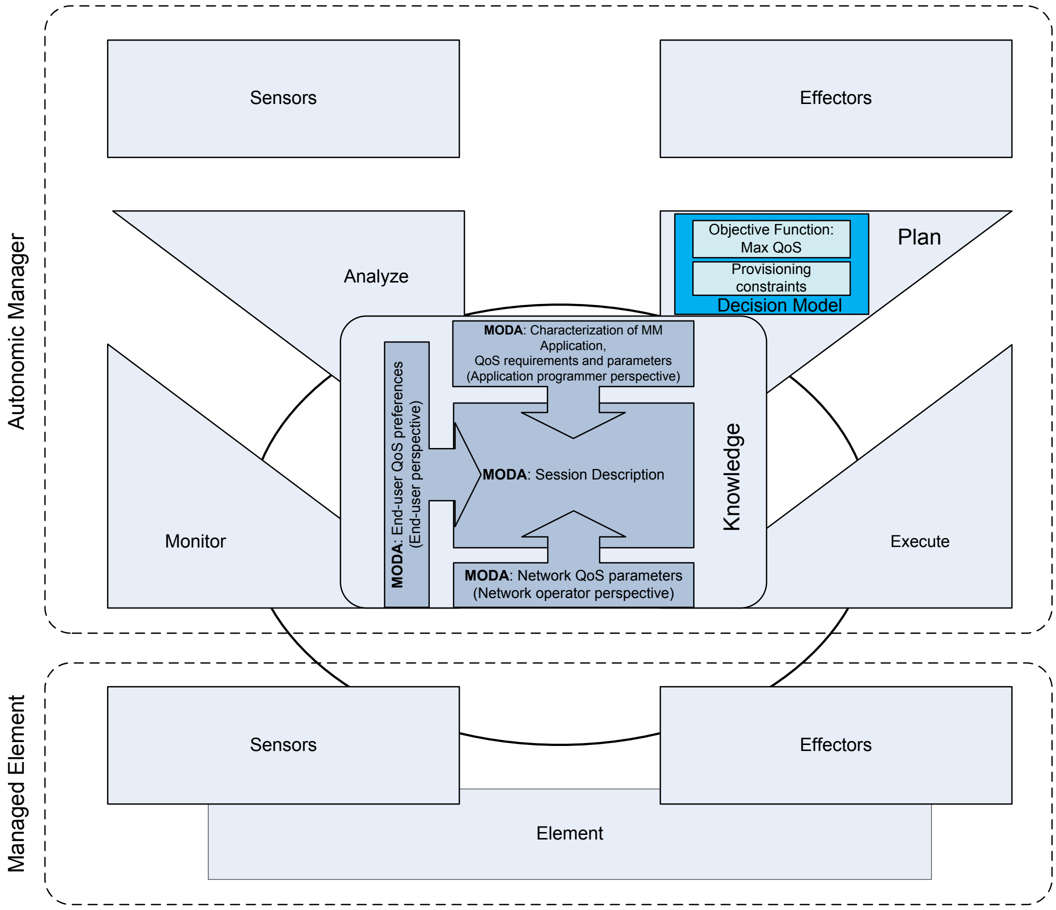

Figure 4.4: Autonomic QoS Management and MODA

As we can see in Figure 4.4, we have based our QoS provisioning model on the Autonomic Computing paradigm and integrating the semantic models of MODA as the source of knowledge. The figure shows the two main components in our work:

- Shared knowledge. The Autonomic QoS Manager has ontology instances as the source of knowledge for each one of the four Monitor-Analyze-Plan-Execute operations in the autonomic control loop. Ontology instances are created from the point of view of end-users, application programmers, and network operator (home network). Also, an instance of the session description ontology based on the information provided by each actors' viewpoint is created.

- Autonomic QoS Management. The autonomic QoS management includes the four MAPE operations. The monitor function collects information (state of the managed element) through the sensor interface. To this end, the managed elements provide a data structure with the information of the device's sources. The analyze function takes such data structure and decides if the autonomic manager can make a decision based on the information provided by MODA. We have focused on the plan operation of the autonomic QoS manager. We have proposed a decision model as part of the plan operation in the autonomic control loop (MAPE). The abstract decision model is defined as a generic model able to be driven by specific home users' configurations. The considered configuration are: global user satisfaction, hierarchical user satisfaction or hybrid (global/hierarchical). Finally, the execute function applies the assignation of network resources through the effector interface.

Figure 4.5 illustrates the interaction between the autonomic QoS manager and its managed devices in the context of a home network. In the figure, MODA is embedded in the autonomic QoS manager.

In order to better understand how MODA and our autonomic QoS framework work in the context of a home network, we have selected the UPnP technology as the mean of communication between the home networked devices. The following section describes the general principles of a UPnP network.

4.3.1 The UPnP Home Network

By visualizing current scenarios where home and small offices' networks interconnect several electronic devices, intelligent appliances, mobile devices, and PCs; the UPnP Forum has emerged as one of the most important initiatives proposing technologies and standards for seamless interconnection and interoperability of IP-based network devices.

Indeed, consumer requirements have evolved and nowadays, user want to transfer video/audio from his media server to TV, control home appliances from his work place, print directly from his camera, and manage every possible home device from his mobile phone or a universal-wide remote control. In order to enable those scenarios, the UPnP Forum has published several specifications through several working committees like, for example, the UPnP Device Architecture, UPnP Audio Video, UPnP Internet Gateway Device, and the UPnP QoS specifications.

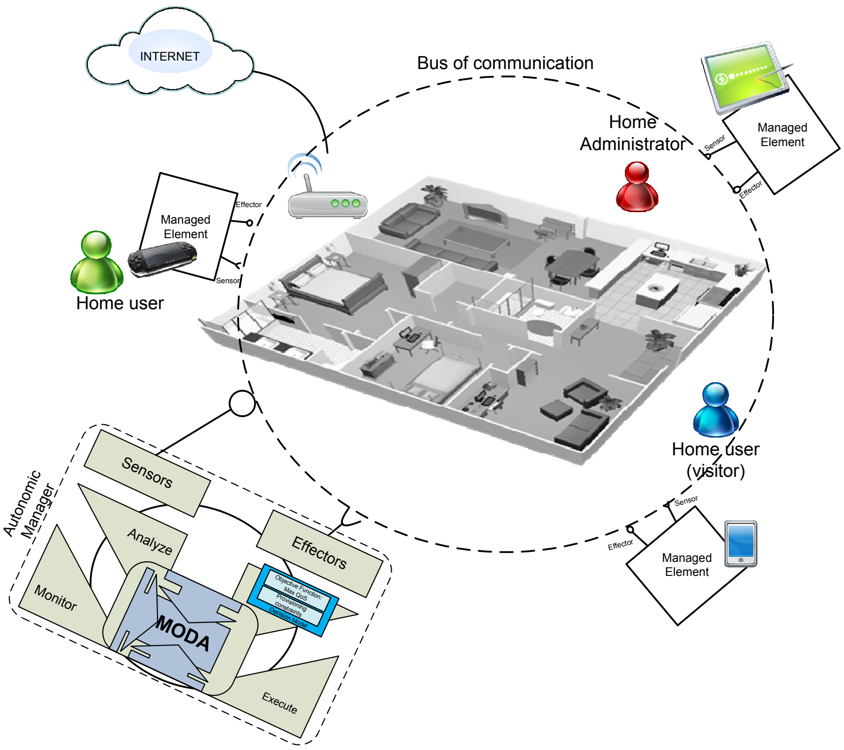

Figure 4.5: Autonomic QoS Management with MODA

UPnP Forum technology is based on the UPnP Device Architecture specification (UDA) [For08b], which is designed to extend the plug and play concepts to network devices and services (i.e. gateways, A/V devices, cameras, telephones, printer, game console, electrical appliances). In order to support zero-configuration, "invisible" networking, and automatic discovering for devices, UDA defines protocols for communication between UPnP control points and controlled devices.

4.3.1.1 UPnP AV

Several works have been carried out in order to deploy and evaluate the UPnP architecture for home networks [GCTD06, LPL08]. In the particular context of multimedia, the UPnP Audio Visual (AV) specification [For08a] has defined a set of UPnP devices and service templates that specifically targets devices interacting with entertainment content such as movies, music, still image, etc.

Three main logical entities constitute the AV architecture: media server, media render, and control point. In the UPnP AV Architecture, the media servers have access to multimedia content and can send it to other UPnP device via the network. Media renderers are able to receive external content from the network and present it on its local hardware. Finally, the control points coordinate the overall operation and provide the interface to the end-user. Figure 4.6 shows the UPnP AV Architecture and the interactions between its components.

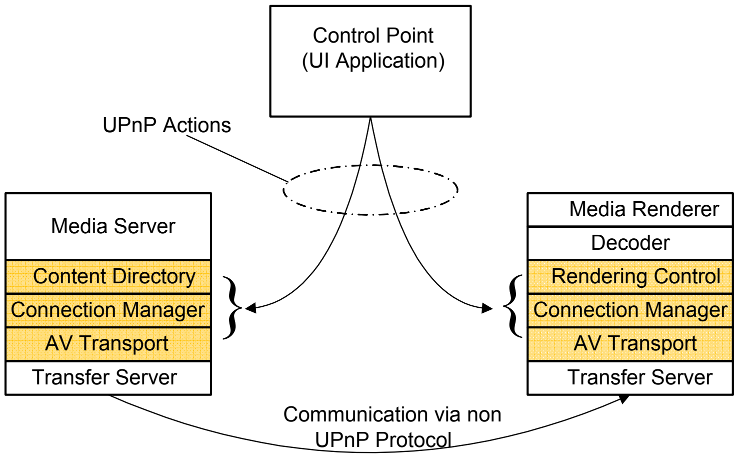

Figure 4.6: The UPnP AV Architecture

The UPnP AV Architecture defines the "three box model" in which the control point is located outside the media server and renderer. The control point could be implemented within a device: either a media server or a media renderer. The UPnP AV Architecture names those cases as the "two box model". No matter the location of the control point, its role and those of the media server and render are well defined by the AV specification. The following paragraphs describe such roles. Figure 4.7 shows a common interaction between the control point, the media server and media renderer in a three box model.

Control point. Usually, a user interacts with a user interface of the AV control point in order to set the multimedia session between the media server and the media renderer. The control point invokes services or actions on UPnP devices (multimedia server and renderer) in order to: browse the multimedia content, obtain a list of transfer protocols and data formats supported by the renderer and server, select a transfer protocol and data format supported by both media server and renderer, configure server and renderer for content transmission, as well as select the item to be transported.

Media server. The media server contains the multimedia content to be browsed by the control point. The media server is composed of three services: the content directory service (CDS), the connection manager service (CM), and the AV transport service (AVT). The CDS service provides the actions that allow the control point to obtain the information about each item that the media server can share in the UPnP network. By using this service, the control point knows meta-information

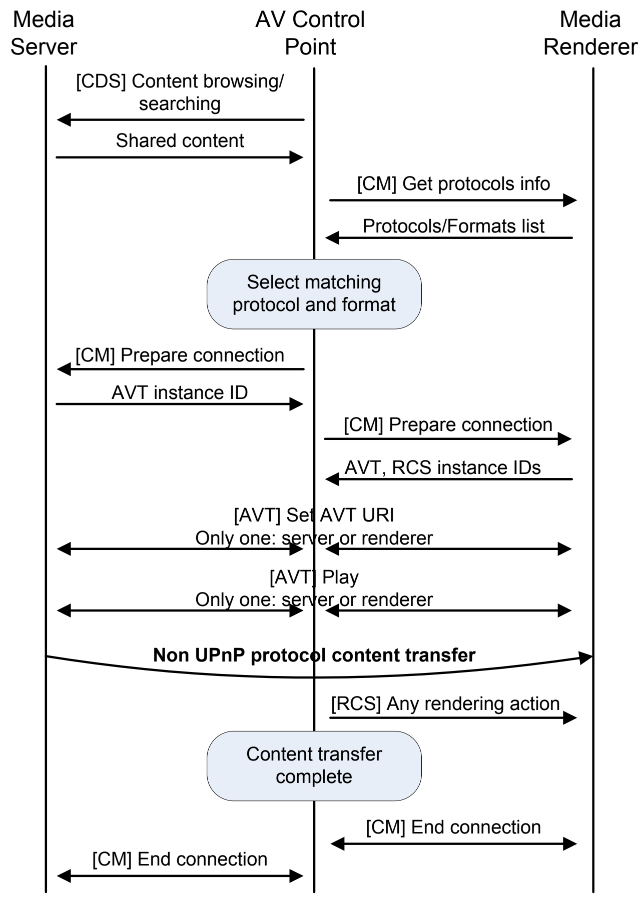

Figure 4.7: Interactions between control point, media server and render in the UPnP AV architecture

about each content item like its name, size, date created, etc. Also, the control point obtains the transport protocols and data format supported by the media server for a particular item. In this way, the control point will know if a media renderer will be able to play that item. Depending on the supported transport protocols and/or data formats, the control point may be able to, e.g., pause, stop, resume, and seek the content that the media server transfers. The AV transport service is an optional service offered by the media server. When implemented, this service allows the media server to distinguish between different instances of this service.

Media render. The media rendered is the device that will play the media content. The media renderer allows the control point to determine the transfer protocol and the data format as well as to control the content flow i.e., play, pause, resume, etc. The media render includes a rendering control service (RCS), a connection manager service, and an AV transport service. The RCS allows the control point to control rendering characteristics such as brightness, contrast, volume, mute, etc.

However, in order to support multiple instances of this service, the connection manager (CM) service must implement the `CM::PrepareForConnection`. The CM service is used to manage the connections with the media renderer. The CM allows the control point to obtain information about transfer protocols and data formats supported by the renderer. In consequence, the control point knows if the renderer will be able to play the selected content item. When the media renderer implements the `CM::PrepareForConnection` action, it allows the control point to control the flow of the content (play, pause, resume, etc.), as well as the rendering characteristics like brightness, volume, mute, etc.

The AV transport service is an optional service used by the control point to control the flow of the content being transferred. In the context of the media renderer, the `CM::PrepareForConnection` enables the creation of several AV transport instance IDs to distinguish between the several connections and instances of this service. In this way, the media renderer can simultaneously handle multiple content items.

Viewed as a Platform Specific Model, UPnP AV session can be modeled using Platform Independent Models of MODA. Indeed, MODA allow taking into account the information utilized by the control point, media render and media server as part of the MODA-based session description (see section 3.4.7.1).

As part of a UPnP Network, the UPnP forum has proposed the UPnP QoS architecture because of the time constraints present in distributed multimedia applications even when using UPnP AV specification. We have mapped our autonomic QoS provisioning approach to the UPnP QoS Architecture. The following section briefly recalls the main functionalities of the services defined in the UPnP QoS architecture in order to better understand the mapping we are proposing.

4.3.1.2 Services of the UPnP QoS Architecture

UPnP QoS Architecture defines three services in order to manage QoS in a UPnP network. The services are: the QoS Policy Holder service, the QoS Manager service, and the QoS Device service. They are defined as follows:

- UPnP QoS policy holder service. It is a repository of QoS policies for the UPnP network. This service provides an interface for a QoS manager to access network QoS policies.

- UPnP QoS manager service. It defines the actions for a control point to setup, release, and update the QoS for a traffic stream. It is responsible for managing QoS assigned to several traffic streams.

- UPnP QoS device service. It is responsible for managing the resources in the device. This service provides an interface for control point to execute actions on the device to admit traffic streams to set up QoS, to query QoS capabilities, to remove QoS of a traffic stream, and to register for events that it generates.

The following section presents the integration of our autonomic QoS provisioning architecture with the UPnP QoS architecture in order to show the feasibility of our approach.

4.3.2 Autonomic QoS management architecture based on MODA

From a designing point of view, this section describes how the autonomic QoS provisioning framework based on MODA could be used and integrated into a particular QoS solution like the UPnP QoS Architecture. The UPnP QoS Architecture was presented in Chapter 2 (see section 2.4.1). Figure 4.8 shows the diagram of the components of the autonomic QoS framework based on MODA used in conjunction with the UPnP QoS architecture. Figure 4.8 also presents the mapping we have established between the autonomic computing components and the UPnP QoS services.

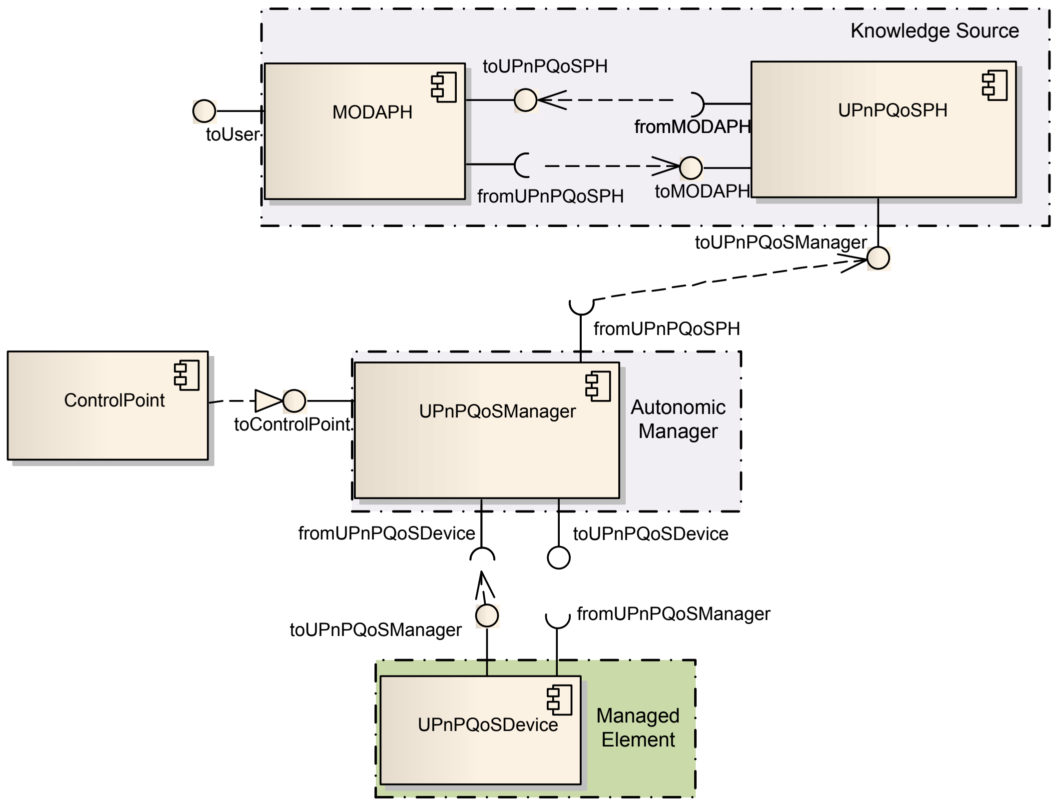

Figure 4.8: Autonomic QoS Management with MODA applied to UPnP QoS

The following sections present our autonomic QoS provisioning architecture from four perspectives. The first perspective is the general presentation from the point of view of the autonomic computing. The second perspective presents the application of our autonomic approach mapped to the UPnP QoS architecture. The third perspective depicts the enforcement of autonomic QoS provisioning from the point of view of the managed element, i.e., from the UPnP QoS device service. Finally, the fourth perspective presents the end-user point of view .

4.3.2.1 Autonomic Computing perspective

From the perspective of the autonomic computing, Figure 4.8 presents the following components:

- `Knowledge source`. This component includes the UPnP QoS Policy Holder service as defined by the UPnP specification and also the MODA Policy Holder (MODAPH) explained here below. As stated in the autonomic computing architecture, the knowledge source provides the information to each MAPE operation of an autonomic manager. The knowledge source is exposed to the user for the definition of the user, application, and network information as well as the definition of decision policies.

- `Autonomic manager`. As we can see in Figure 4.8, the autonomic manager operates on the managed element (UPnP QoS device) through its effector interface and also it takes the necessary information from the knowledge source in order to make decisions when a new request arrives (from the control point). The decisions of the autonomic manager are based on a utility function and IF-THEN actions defined in the Plan function of its MAPE control loop.

- `Managed Element`. This component includes the UPnP QoS Device service as defined by the UPnP specification. As stated in the autonomic computing architecture, the managed element is controlled by the autonomic manager through its effector interface defined in conjunction with the UPnP QoS device service. Certainly, the managed element should implement all the functions to actually enforce the QoS in the device.

4.3.2.2 UPnP QoS Architecture perspective

From the perspective of the UPnP QoS architecture, the following paragraphs detail each one of its components:

- `ControlPoint`. The control point is the component that starts the QoS setup process. By using a traffic descriptor structure, the control point is the entity that, at the beginning, asks the UPnPQoSManager to setup a certain QoS for a traffic flow.

- MODAPH. This component deals with the Multimedia Ontology Driven Architecture (MODA) framework. MODA framework includes a set of ontologies which allows the user to express his QoS preferences for other users and/or multimedia applications. The MODAPH component also offers a graphic user interface through which the user communicates his preferences. When necessary, the MODAPH component communicates with the UPnPQoSPH component in order to provide a *traffic importance number* and a *user importance number*. Finally, MODAPH component deals with all the ontology treatment i.e., reasoning, storage, processing, creation [EE09, EGL09, EGM10, GMEL09, GMLE09, GME10].

- UPnPQoSPH. This component has been defined in the UPnP QoS Architecture. The main purpose of this component is to provide the policy for a given AV flow. When requested by the UPnPQoSManager, the UPnPQoSPH returns the traffic importance number and the user importance number for a given flow. The UPnPQoSPH can send requests to the MODAPH in order to apply more complex QoS policies, for example, QoS policies aimed at groups or combinations of users and applications.

- UPnPQoSManager. The UPnPQoSManager is requested by the control point in order to setup, release, or update the QoS for a traffic flow. In a prioritized QoS UPnP scenario, the UPnPQoSManager requests the UPnPQoSPH to provide the traffic importance number for a given flow. The UPnPQoSManager will also interact with all the UPnPQoSDevices in the end-to-end path, in order to setup and admit the requested QoS (effector interface). In a parameterized QoS UPnP scenario, the UPnPQoSManager will request every UPnPQoSDevice in the end-to-end path in order to reserve, for each segment, the required network resources. Also, the UPnPQoSManager will request the UPnPQoSPH only if preemption is demanded by the control point; it means that the UPnPQoSPH will return the importance of the existing traffic flows and then, the UPnPQoSManager will preempt the flows with lower importance and, as result, it will admit the more important ones. A hybrid QoS UPnP scenario includes segments where prioritized QoS is not possible and then, for such segments, parameterized QoS will be applied.

- UPnPQoSDevice. The UPnPQoSDevice is responsible to manage the resources within the UPnP device. This component offers, through its sensor interface, the actions that allow, for example, knowing the QoS capabilities of the UPnP QoS device, the state of the UPnPQoSDevice's service as well as to admit a QoS request, to release resources, to update resources requests, among others actions. This component also allows knowing the UPnP device network parameters like its network interfaces, maximum bandwidth, etc. Another important function of this component is to map information like the traffic importance number or the information contained in the traffic descriptor to a layer 2 class of service through its effector interface. In this sense, when prioritized QoS is requested by the control point, the `UPnPQoSDevice` service will always accept the traffic stream with its respective priority (effector interface). Whereas in a parameterized QoS segment, the `UPnPQoSDevice` must reserve the solicited resources for a given flow and if this is not possible, it must notify this to the `UPnPQoSManager` (sensor interface).

4.3.2.3 Service management perspective

The `UPnPQoSDevice` service is, actually, the responsible for the enforcement of the QoS defined by the UPnP QoS manager. From a programming point of view, we have specially focused on the `UPnPQoSDevice` Service. This is the reason why we explain in more detail the `UPnPQoSDevice` service. Figure 4.9 shows a class diagram of the `UPnPQoSDevice` service.

We propose the `UPnPQoSDevice` service be organized in four packages: `Action`, `Device`, `StateVariable`, and `Tools`. The `Action` package contains the 10 actions defined as obligatory in the UPnP QoS Architecture specification. The `Device` package contains the implementation of the `UPnPQoSDevice` service as an UPnP device. The `Device` package contains the `Activator`, `QoSDevice`, and `QoSDeviceService` classes. The `StateVariable` package includes all the classes representing the UPnP QoS Device state variables. There are 22 mandatory state variables. Finally, the `Tools` package includes all the classes used to accomplish the utility tasks of the `UPnPQoSDevice` service, for example, the XML management or the classes in order to know the MAC addresses are some of the tasks done by classes in this package.

Figure 4.10 illustrates the sequence diagram for setting the required QoS initiated by the *RequestTrafficQoS* message sent by the Control Point to the UPnP QoS Manager. The QoS Manager collects information about the different QoS Devices in the network. Actually, in order to obtain the device-to-device path information, the `UPnPQoSManager` uses the *GetPathInformation* method of each UPnP QoS device in the device-to-device path.

The Policy Holder provides the *TrafficImportanceNumber* (TIN), which is obtained by calling the *GetTrafficPolicy* method of the `UPnPQoSPolicyHolder` service. Once the UPnP QoS Manager has received the policy information for a given flow, it communicates with the `UPnPQoSDevice` service in order to setup

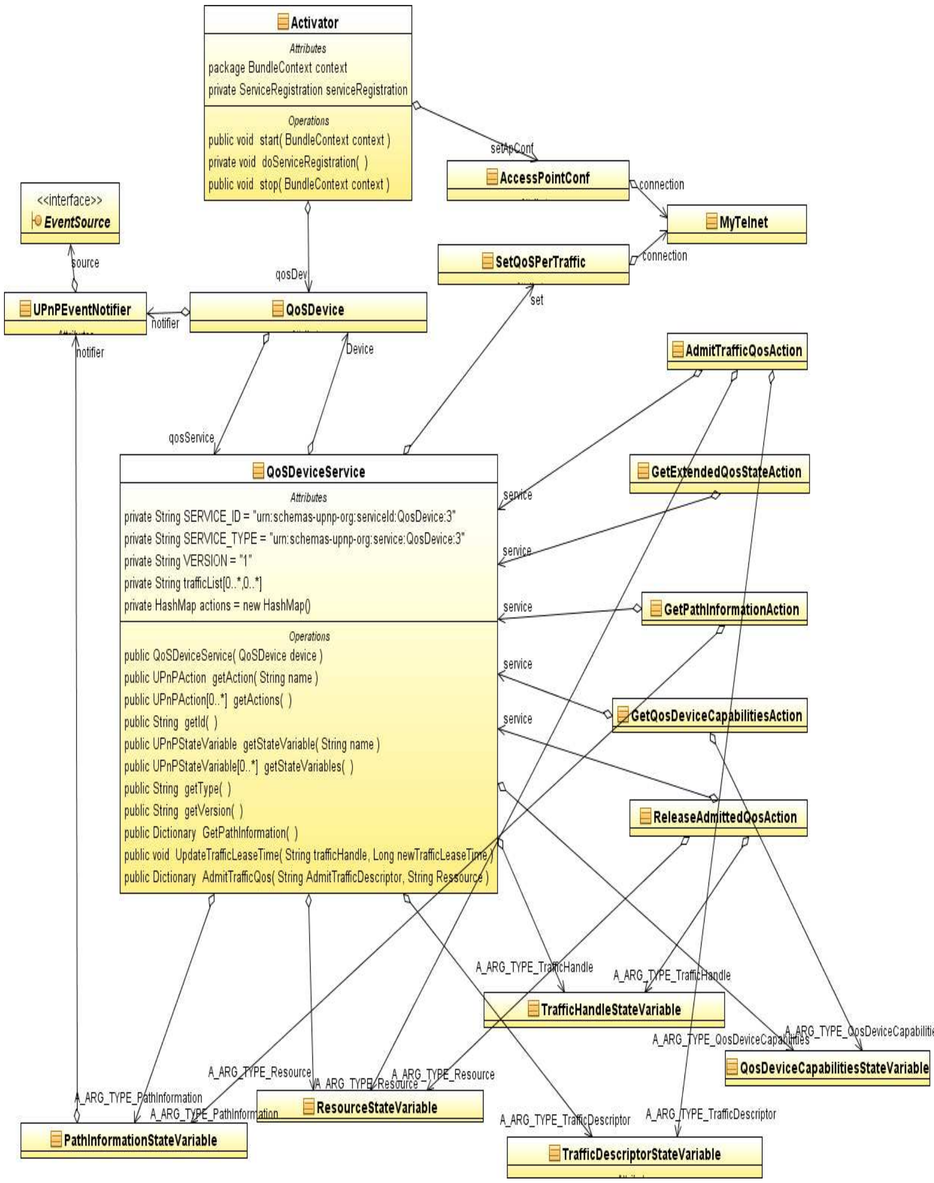

Figure 4.9: Class diagram of the UPnPQoSDevice service

the required QoS by invoking the *AdmitTrafficQoS* method of the `UPnPQoSDevice` service. The `UPnPQoSManager` service provides an XML file with a *TrafficDescriptor* structure containing the *TrafficImportanceNumber*, *ports*, and *IP addresses* among other information.

The `UPnPQoSDevice` service uses the information transported in the *TrafficDescriptor* in order to configure the required QoS in the Access Point. Such configuration is done by calling the *SetConfAccessPoint* method.

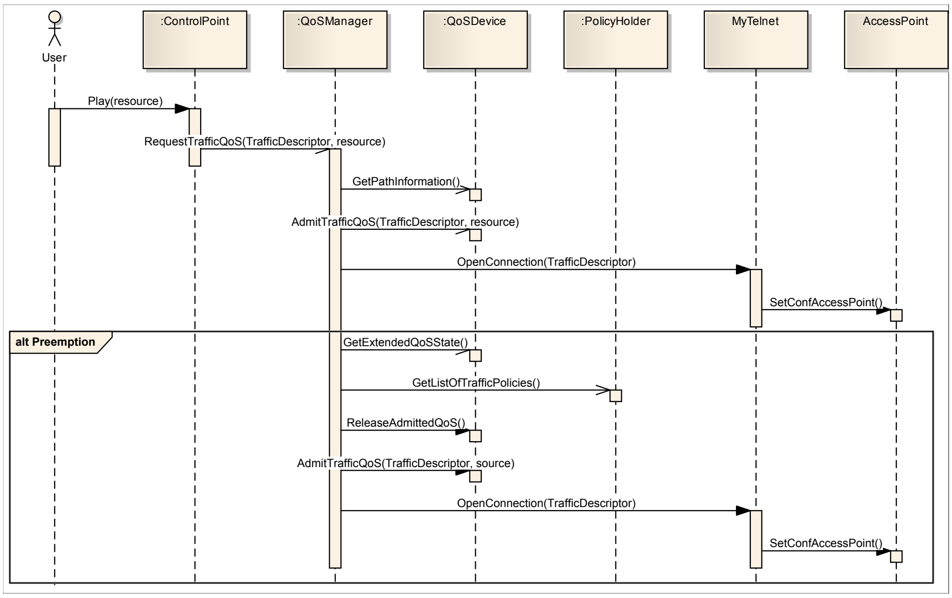

Figure 4.10: Sequence diagram for setting the QoS in the UPnP QoS Architecture with MODA

Another important activity in the process of setting the QoS by the Autonomic QoS Manager is the establishment of the necessary QoS in all the path from the source device to the destination device in the home network. To do this, the UPnP QoS Manager gathers the information of the different UPnP QoS Devices in the home network. The `UPnPQoSDevice` service responds to the *GetPathInformation* method invoked by the `UPnPQoSManager` service in each UPnP QoS device. Figure 4.11 shows the sequence diagram to obtain all the QoS Devices in the communication path.

4.3.2.4 User perspective

In our framework for QoS provisioning, user preferences are taken into account in the decisions of QoS management. In order to allow home users to easily participate in the QoS management, a friendly user interface has to be provided. Based on the ontologies included in MODA, a graphical user interface (GUI) can be automatically generated (e.g., using ontology-based frameworks such as Protégé[2]). This GUI is built based on all the possible combinations of multimedia services that can be deployed at home. Examples of these services are Video on Demand (VoD) services described as sending tasks, or conversing and conferencing applications such as MSN or Skype. Figure 4.12 illustrates the process of automatic GUI

[2]http://protege.stanford.edu/

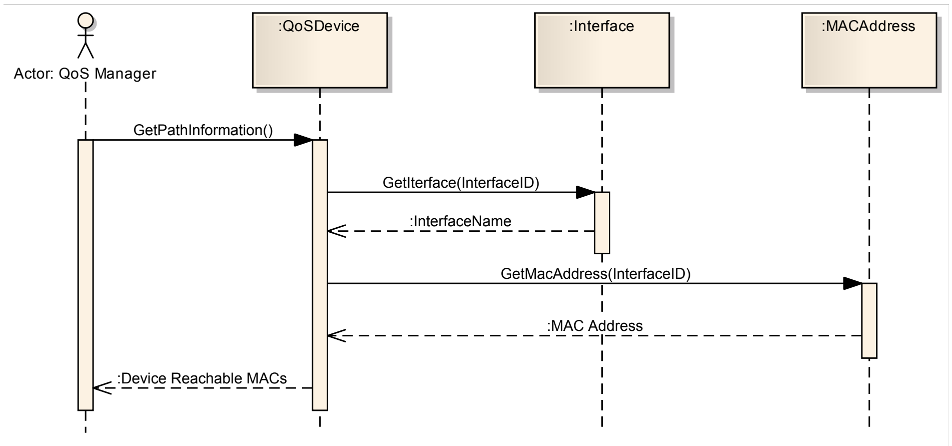

Figure 4.11: Sequence diagram for the determination of the device-to-device path

interfaces generation based on the ontologies of MODA and its storage within the QoS policy holder.

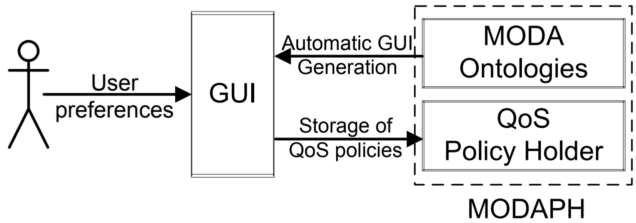

Figure 4.12: Collecting user preferences for autonomous QoS management

Based on the semantic models of MODA, actors are able to instance its corresponding semantic model of MODA. For example, end users can provide its preferences for all the applications by specifically selecting a communication task or any application instance represented by an application task. This list of multimedia application patterns or specific application instances will be available to the home users in order to express their preferences. Any user can provide these preferences by granting a numeric priority between the applications. For the administrator user (i.e., any authorized home user owning administration rights), additional priorities can be granted to the home users. The user and application priorities will be automatically stored within the `UPnPQoSPolicyHolder` service. Once that user has provided his QoS preferences (user and application priorities), they can be considered for assigning network resources.

Figure 4.13 illustrates the Protegé's graphic user interface allowing an application programmer to define the composition of a distributed multimedia system at design time.

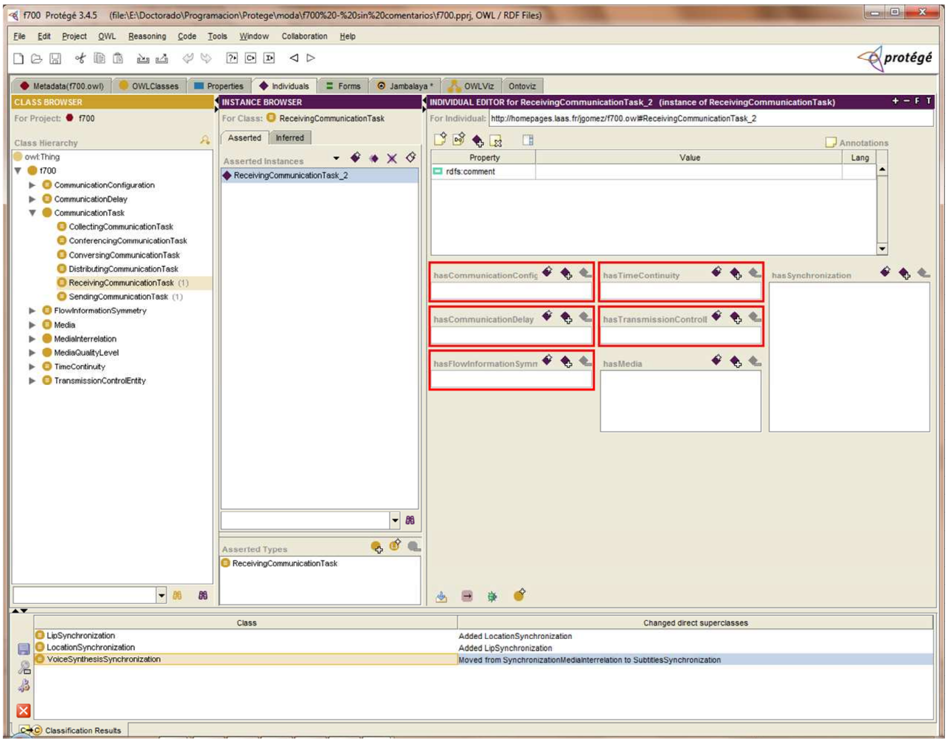

Figure 4.13: Protege's GUI for an application programmer to design a DMS composition

4.3.3 Summary

This section has presented the Autonomic QoS management architecture based on MODA. In the first part of the section, we have presented the design of the QoS provisioning framework following the Autonomic Computing principles. Particularly, we have focused on the Autonomic QoS Manager and its integration with the semantic model of MODA applied to UPnP home networks. Also, a brief explanation of UPnP networks has been presented.

The second part of the section has presented the mapping between the components of our autonomic QoS provisioning framework and the components of the UPnP QoS architecture. Also, we have shown the design of the UPnP QoS Device service which is the managed element in the enforcement of the required QoS. A class diagram showing the structure of this service has been presented as well as the main sequence diagrams that illustrate the interaction of the components of the autonomic QoS framework in order to set the QoS parameters along the device-to-device path have been presented.

4.4 Chapter Conclusion

The first part of this chapter has presented the state of art of the Autonomic Computing and Communications paradigms. We have seen that the properties of the autonomic approach are well suited to our objective: to provide QoS for distributed multimedia systems, considering both users and applications preferences and requirements in the context of home networks.

The main reason that guide us to make this choice is that Autonomic Computing facilitates management tasks to humans. This is particularly important for QoS provisioning of multimedia applications in the context of home networks. Autonomic Computing allows us to design a framework that automatically manage network resources driven by the preferences of non-expert users. We have focused on the design of an Autonomic QoS Manager able to make autonomic decisions as part of its Plan operation in the MAPE control loop for self-configuring the network resources.

The second part of this chapter has introduced our Autonomic QoS Management Architecture based on MODA. MODA is integrated in the Autonomic QoS Management Architecture as the knowledge base. All the information provided by the semantic models of MODA is used by the Autonomic QoS Manager in the MAPE operations.

The autonomic QoS management architecture has been presented from four viewpoints. The first point of view presents the general QoS management architecture from the autonomic computing perspective. The second viewpoint presents the integration of our autonomic approach with the UPnP QoS architecture. The mapping between the components of both architectures autonomic and UPnP QoS is depicted by using a component diagram. The third perspective takes into account the viewpoint of the managed element, i.e. the UPnP QoS device which is actually the entity that enforces the QoS. Finally, the fourth perspective presents the process through which the actors interact with MODA and it also presents a GUI for an application programmer to design the patterns of DMS.

The following Chapter 5 shows how we have developed the semantic models of MODA and the Autonomic QoS Manager in the context of the Feel@Home project.

CHAPTER 5

Deployment and Evaluation

Contents

5.1 Introduction

THE previous chapters have introduced the two main contributions of this thesis: Multimedia Ontology Driven Architecture and Autonomic QoS management framework. This chapter is intended to describe how these contributions have been deployed and evaluated in the framework of the Feel@Home project.

This chapter is organized in two main sections. The first section presents the Feel@Home project and describes the main QoS scenarios where our contributions have been developed and evaluated. The second section shows how to use MODA by characterizing a session of a Video-on-Demand application in the context of a home network. A generic decision model as part of our proposed autonomic QoS manager is also presented. The generic decision model takes semantic information from MODA in order to manage the provisioning of network resources. This section also presents how the QoS is enforced from a device point of view, thus it presents how the self-configuration is done at the level of the QoS device. Finally, this section presents the experiments carried out in order to validate the services offered by the autonomic QoS manager, as the complete autonomic QoS management architecture based on the knowledge source provided by the MODA semantic models.

5.2 The Feel at Home Project

The Extended Home concept deals with scenarios where home-based services are accessible to home users whether they are at home or outside. In other words, the user has access to his home services from his car, office, or other homes and also he has similar experience of such services no matter his location. Projects like Feel@Home [BMMMMD11] have targeted and developed such kind of scenarios.

In the context of an extended home, new ubiquitous multimedia scenarios take place. In these scenarios, the home user is able to run multimedia applications whether the user is at home or not. Furthermore, the user has the means to express his preferences (e.g. QoS preferences) for the operation of the distributed multimedia services. Certainly, frameworks that take into account user's preferences for the spontaneous construction, configuration, and deployment of distributed multimedia sessions are needed.

This thesis has been developed in the framework of the Feel@Home project. The goal of the project is "to build a system able to make applications capable of accessing a personal multimedia contents stored at home, both anytime and anywhere". Among the functional requirements of the project, one can find the following:

- To access home multimedia contents while being at home and from outside home
- To share multimedia contents with other users
- To access different services and
- To share services

Among the non-functional requirements addressed by the Feel@Home project one can find:

- To secure communication in order to avoid misuse of confidential data and prevent the access of non-allowed users
- To provide QoS in order to offer a good Quality of Experience in both cases: being intra-home and from outside the home.

As we mentioned in the introduction of this thesis, we have focused in the QoS management based on user preferences when the user is in his home (home network). In other words, we are focused on the QoS management provisioning, a non-functional requirement, of the Feel@Home project.

In the Feel@Home project, specifically, for the home networks, the UPnP technology is used. This means that the multimedia contents will be automatically discovered by using the UPnP protocols. Also, for this thesis, we have selected the UPnP technology as the communication technology in the home network for testing purposes.

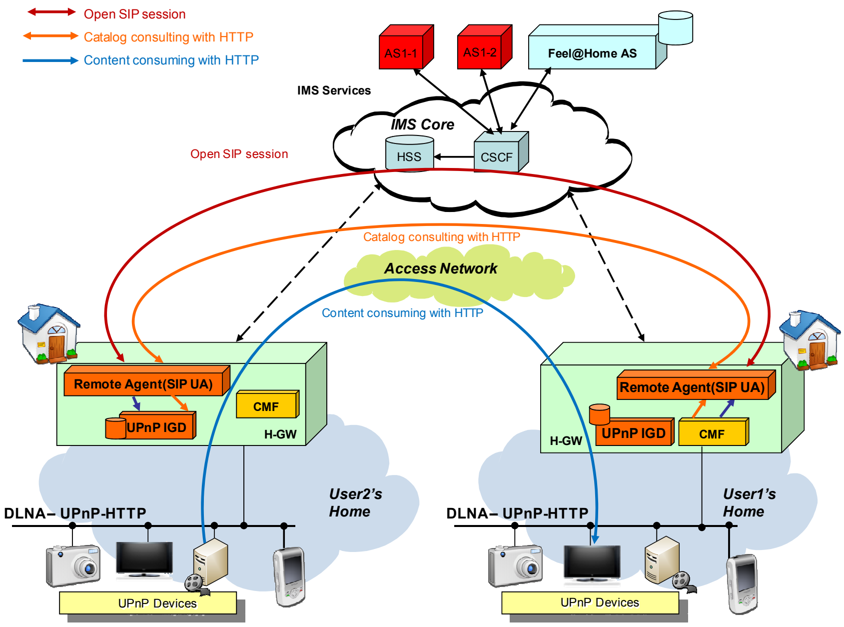

Figure 5.1: Feel@Home architecture overview (taken from [BMMMMD11])

Figure 5.1 shows an overview of the architecture of the Feel@Home system when the remote distribution of the media contents is done through IP Multimedia Subsystem (IMS). According to [BMMMMD11], Figure 5.1 depicts the case in which two users want to share their contents. When using IMS, the Feel@Home architecture architecture includes the following components:

- a compliant remote agent with the SIP IMS user agent behavior.
- a UPnP IGD to manage network address translation (NAT) and firewall rules.
- a Cross Media Finder (CMF) that is the equivalent of a content control point.
- a Feel@Home Application Server.

5.3 Deployment and Evaluation

This section begins by describing the utilization of the semantic model of MODA. It continues by presenting the decision model used by the autonomic QoS manager and its evaluation. After this, it is presented the actual enforcement of the QoS at device level. To this end, the auto-configuration procedure is presented. At the end of the section, the executed experiments and results are described.

5.3.1 Policy Holder and Knowledge source: Instantiation of the models in MODA

This section describes how the semantic models of MODA are used to characterize non-functional information of multimedia applications in the context of a home network. For this purpose, in the first part of this section, we present the characterization of a Video-on-Demand application. This example will allow us to show how ontologies of MODA are able to create a semantic map of the existent flows in the home network. The second part of this subsection illustrates how QoS can be autonomically provided by managing the resources of home network communication devices based on the semantic models of MODA. We have developed a test scenario for which it has been necessary the JAVA implementation of the basic functionalities proposed by the UPnP QoS Architecture.

Use of Semantic Models of MODA. In the Feel@Home project, there is a widely used scenario in which a user wants to watch a film or listen to a song by easily selecting the content from a media server and choosing the device (render) where the content will be played with the help of a remote control. For our testing purposes, let's suppose that a home user wants to execute a Video on Demand (VoD) application. He starts the VoD from his mobile phone using the Wi-Fi connection from his home network. Both the user and the VoD have a priority value assigned for QoS provision purposes. Also, the VoD application should have predefined QoS requirements. Such information can also be used by the UPnP QoS Policy Holder and the UPnP QoS Manager like the UPnP QoS Architecture proposes for the QoS provision. Let's see how the MODA framework is used to characterize this scenario.

First, the semantic description of the VoD application is achieved by instantiating the semantic model of the ITU-T Rec. F.700. The characterization of the VoD application includes the following elements:

- Communication Configuration. The ontology allows only a point-to-point configuration to be established because it is a sending task. With this information, we can infer that there is a source and sink host (both of them having parameters like IP address and ports).

- Communication Delay. Since the VoD is a sending task, the system graphic user interface (GUI) enables the options: near-real-time, real-time, non-real-time, specified-time. It means that the ontology provides the options: "wait a few seconds for having a better quality" (near-real-time); "start immediately" (real-time); or "watch it 2hrs from now" (specified-time).

- Symmetry of information flow. As it is a sending task, VoD has unidirectional symmetry of information flow. This means the audio and video will be sent from source to sink.

- Transmission Control Entity. The only option is "source control" because of the definition of sending communication task.

- Time continuity. It is related to communication delay thus, for instance, for a VoD service the appropriate configuration would be an Non-isochronous transmission (buffering capable) with a near-real-time delay.

- Media. According to ITU-T Rec. F.700, audio and video may be used. Besides the ontology allows setting the quality level of both audio and video media, e.g., "a speech quality", equivalent to an A3 quality level (MPEG4 codec), and "High definition video quality", equivalent to a V4 quality level (High Definition video quality).

- Media Interrelation. Within a VoD application with audio and video flows there is a relationship of synchronization between them. By using the ontology, it is possible to configure the type of synchronization e.g. lips synchronization (audio and video flows), or subtitles synchronization (video and text), or both of them lips and subtitles synchronization (audio, video and text).

Figure 5.2 shows the graph representation of the VoD individual already configured.

For the VoD example, we have defined two QoS requirements based on the ontology of the ITU-T Recommendation X.641 (see Figure 5.3):

- Packet Loss Ratio (PLR). We define the packet loss ratio requirement as recommended in ITU-T Recommendation G.1010 [Uni01b]. This requirement is measured as a percentage. The QoS parameters of PLR are the *Controlled-HighestQuality* with a value of 0.0 meaning 0% of loss; and the *Lowest-QualityAcceptable* which has a value of 1.0 meaning 1% of loss. Semantically speaking, the lower the parameter value, the better the QoS. Finally, its *AgreementLevel* is *compulsory* meaning that the application must execute the necessary actions to recover an acceptable QoS level when this requirement is no longer accomplished.

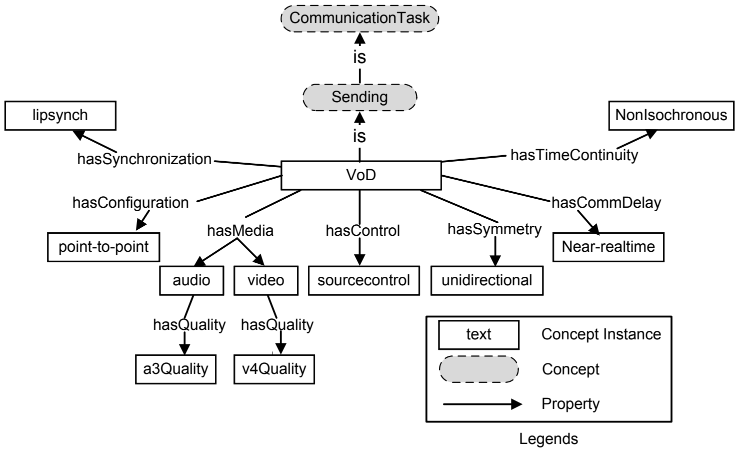

Figure 5.2: ITU-T F.700 ontology instance for a VoD Application

- One-way delay. This requirement is measured in seconds. According to ITU-T Recommendation G.1010, the *ControlledHighestQuality* parameter value is 0.15s and the *LowestQualityAcceptable* values is 10s of delay. Here, this requirement must be *guaranteed*.

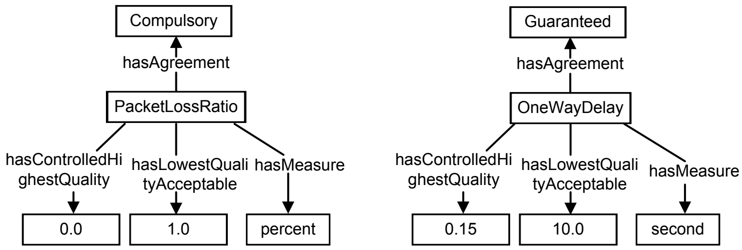

Figure 5.3: Semantic characterization of the PLR and One-Way delay QoS requirements

MODA offers a GUI through which users can set their multimedia application and priorities as follows (see Figure 5.4):

- User. An individual of user class has a name John Doe. John Doe has the username jdoe, and the email account john.doe@myhome.com. Also John Doe has a priority of 255 as the highest priority.

- User role. Given that John Doe owns the home network, the ontology GUI allows him to set his role as an Administrator user.

- Application. The user can select one of the applications previously instantiated (characterized) using the ontology of the ITU-T Rec. F700, e.g., the

VoD application. A user may assign a priority to an application with values in the range of 0 through 7.

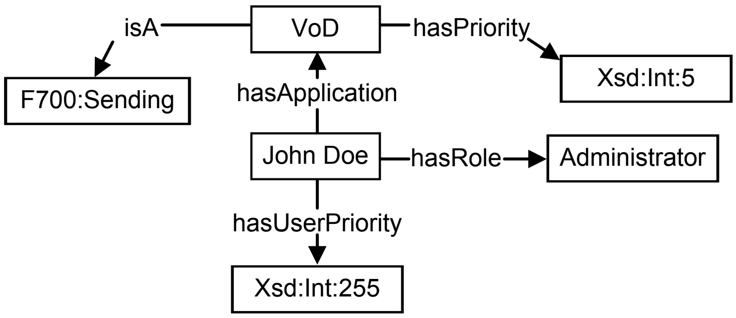

Figure 5.4: Characterization of user and application priorities

In order to characterize the device from which the home user execute the multimedia application, an instance of the Host-Network ontology is created as follows (see Figure 5.5) :

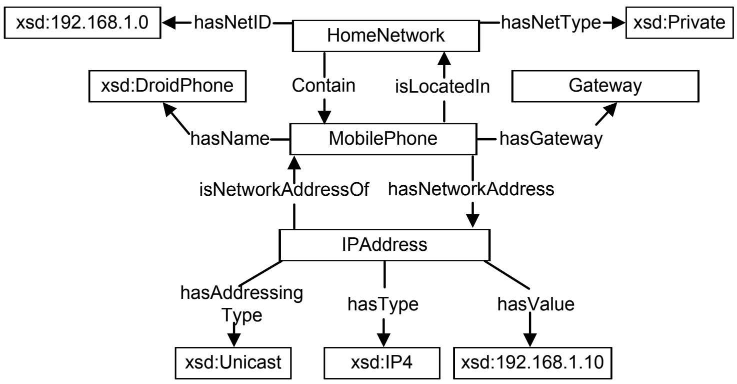

Figure 5.5: Characterization of user's device

- Host. According to the scenario, the instance of the concept is a *MobilePhone*. It has a *Gateway*, which is also a host and then it is characterized. The *MobilePhone* is located in a *HomeNetwork* and it has an *IPAddress*.

- Network. The concept helps to characterize the *HomeNetwork* which contains the *MobilePhone*. The home network has a network type which is *Private*. Also, it has a network ID, which corresponds to 192.168.1.0.

- Network Address. The concept allow the characterization of the Internet address of the device in this case an *IPAddress*. It has a *Unicast* addressing type. The network address type is *IP4*, which has a value of 192.168.1.10 assigned to the *MobilePhone*.

Once the multimedia service and its information are characterized, the MODA framework allows generating the session description by instantiating the proposed session description ontology as depicted in the Figure 5.6.

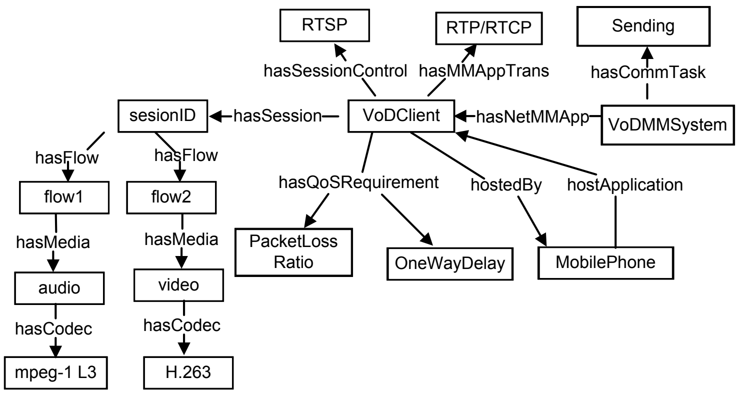

Figure 5.6: Characterization of the session by means of MODA's ontologies

The session description ontology of the VoD application makes reference to its QoS requirements (ITU-T Rec. X.641), Host-Network, multimedia task (ITU-T Rec. F700), and user-application priorities ontologies. The QoS requirements are based on the ITU-T Rec. G.1010. Indeed, characterization of QoS requirements helps us to apply QoS policies when QoS requirements are no longer achieved. As a consequence, appropriate actions can be triggered to fix or alert potential problems.

The following subsection presents the enforcement of the QoS done by our autonomic QoS manager, which takes the semantic models of MODA as the source of knowledge. The autonomic QoS provisioning framework has been applied to the UPnP QoS Architecture in the context of a home network. Here, we firstly present how the autonomic QoS manager works.

5.3.2 Autonomic QoS Manager: Generic Decision Model based on MODA

The proposed autonomic QoS manager includes a decision model driven by an objective function representing the degree of satisfaction of the QoS. Two scenarios can be considered to define QoS satisfaction. The first scenario describes a global QoS satisfaction, where the optimal provisioning solution is the one offering the best compromise between the resources usage and the priorities between users and applications. A second scenario extends the global satisfaction scenario by adding preemption for high priority traffic. Priority information is taken from the corresponding ontology instance of MODA. The generic QoS provisioning model is defined as follows:

Let $T = \{t_1, \dots, t_n\}$ be the set of n traffic streams contesting for the available services. Also, Let $S = \{s_1, \dots, s_m\}$ be the set of m classes of services.

Let $x_{ij} = \{0, 1\}$ be the decision variable associated to the use of the service s_j to transmit the traffic stream t_i. Traffic stream information is taken from the ontology-based session description.

Let a_j be the maximum availability of service s_j. Let $p_i = \{1, \dots, p_{max}\}$ be the priority associated to the traffic t_i.

Let c_i be the cost of the traffic stream t_i (for instance the bandwidth required for t_i as described in its Traffic Specification (TSPEC) and session description). Then, our generic decision model is defined as follows:

$$max\, Z = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \times x_{ij} \tag{5.1}$$

subject to

$$\sum_{j=1}^{m} c_i \times x_{ij} \leq a_j, \forall j \in S \tag{5.2}$$

$$\sum_{j=1}^{m} x_{ij} \leq 1, \forall i \in T \tag{5.3}$$

$$x_{ij} \in \{0, 1\},\ \forall i \in T,\ \forall j \in S \tag{5.4}$$

The equation 5.1 defines the objective function for global or hierarchical user satisfaction. In this function, a key parameter is represented by the priority coefficient. Indeed, this coefficient can be adapted in order to provide global or preemptive QoS satisfaction:

- For global satisfaction p_i is defined as the multiplication of user and application (i.e., traffic flow) priorities. In this case, the best possible s_j resource assignation will be computed.
- For preemptive scenarios, new constraints need to be added in order to allow a global satisfaction of users but respecting mandatory traffic priorities. This is achieved by including constraints such as: $\sum x_{kj} = 1$, meaning that for the prioritized traffic t_k enough resources are always reserved, and only the remaining resources will be employed for global satisfaction.

The constraint 5.2 is used for admission control purposes, based on the cost of the traffic streams and limited to the maximum of resources. The constraint 5.3 is included in order to limit the satisfaction of a traffic stream to a unique class of service. It means that a traffic t_i can be assigned to only one class of service.

The proposed generic model decision allows to consider other factors when calculating the best solution. For example:

- Feedback of user experience: QoS provisioning models need to learn from users' satisfaction in order to improve the decision process. In our decision model, this is achieved by adjusting the p_i coefficients when the user feedback is not satisfactory, i.e., based on QoE. For instance, when the user express his feedback using the ontology-based GUI, the adequate user and application priority will be adjusted at the policy holder level in order to accurately compute the corresponding p_i coefficient for future decisions.

- Service compatibility: If a service s_p is not compatible with a traffic t_r, the following constraint could be added in order to avoid the use of this service: $x_{rp} = 0$

In order to put into practice the decision model, we have developed an scenario in which multimedia and non multimedia flows are transmitted. The decision model assigns network resources while considering *user* and *traffic priorities*. The following section details such scenario.

5.3.2.1 Proof of concept

A study case has been elaborated in order to evaluate the feasibility and the advantages offered by the decision model. The following elements have been defined:

- Users: local users with high or normal level priorities as well as home-visiting users with low priorities have been included. These priorities define the p_i set.

- Applications: several multimedia applications (i.e., Audio/Video on demand, audio/video conversing) as well as other standard distributed applications (i.e., file downloading) have been simulated. Bandwidth constraints of the various multimedia codecs have been used to define the cost (c_i) set.

- Classes of services: guaranteed bandwidth services of 10Mbps as well as best effort services have been defined. In this study case, only the global satisfaction scenario will be studied.

- Decision algorithm: the decision manager has been implemented by using research operation algorithms.

Table 5.1 summarizes the results of the simulation of 15 streams. The table also includes the associated costs in Kbps. The simulation was performed in two phases. During the first phase the global satisfaction was of Z=550 and 7 streams where included in the solution (i.e., the guaranteed service was assigned to the streams).

			Phase 1 (Z=550)		Phase 2 (Z=440)	
ID	**Application**	**Cost**	**Priority**	**Solution**	**Priority**	**Solution**
T1	VoD hd	6000	100	Accepted	100	Accepted
T2	VoD hd	6000	50	Not Accepted	50	Not Accepted
T3	VoD sd	1600	100	Accepted	100	Accepted
T4	VoD sd	1600	50	Accepted	50	Not Accepted
T5	VoD sd	1600	10	Not Accepted	10	Not Accepted
T6	AoD hd	384	10	Not Accepted	10	Accepted
T7	AoD sd	128	100	Accepted	20	Accepted
T8	AoD sd	128	50	Accepted	20	Accepted
T9	AoD sd	128	10	Not Accepted	20	Accepted
T10	VoIP	64	100	Accepted	100	Accepted
T11	VoIP	64	10	Not Accepted	10	Accepted
T12	Videophony	448	50	Accepted	50	Accepted
T13	Videophony	448	10	Not Accepted	10	Accepted
T14	file transfer	VBR	100	Not Accepted	100	Not Accepted
T15	file transfer	VBR	50	Not Accepted	50	Not Accepted

Table 5.1: Simulation results

The second phase corresponds to an adaptation of the traffic priorities for T7, T8 and T9. Indeed, the audio on demand application producing the T9 stream has been in competition with other applications within the best effort service, and the QoS obtained was not satisfactory. The home administrator has modified the policies assigned to this kind of applications and the new priority for any user running this application is fixed to 20. A new assignation of resources has been computed, and even if the global satisfaction has been reduced to Z=440, audio on demand applications has been included in the guaranteed service. In this second phase, 10 streams were included in the solution.

This basic scenario illustrates the feasibility of the decision manager to assign network resources while performing global user satisfaction. However, this study also raises the complexity involved in defining priorities for users and applications. Certainly, the proposed MODA framework can help users to translate their requirements and preferences by instantiating its semantic models and thus aligning users preferences (QoE) and application QoS requirements in order to manage home resources utilization. The following section details the QoS enforcement.

5.3.3 Managed Element: autonomic enforcement of the QoS (self-configuring)

In the UPnP QoS Architecture, the QoS devices (Managed Elements) are responsible to enforce the decisions taken by the QoS managers (Autonomic Manager) in order to offer the most adequate QoS to the final users. The way this QoS enforcement is achieved depends on the type of devices and the mechanisms that are available to manage their network resources. In order to illustrate the QoS enforcement process, an access point compliant with the UPnP QoS specification is presented. This access point is able to manage QoS by defining access control lists in order to accept specific traffic classes.

```
ap(config)#access-list 101 permit udp any any precedence 1
ap(config)#access-list 102 permit tcp any any precedence 2
ap(config)#access-list 103 permit udp any any precedence 0
ap(config)#access-list 104 permit tcp any any precedence 3
ap(config)#access-list 105 permit UDP any any precedence 4
ap(config)#access-list 106 permit udp any any precedence 5
ap(config)#access-list 107 permit UDP any any precedence 6
ap(config)#access-list 108 permit udp any any precedence 7
ap(config)#access-list 109 permit udp any any eq 10000
ap(config)#access-list 110 permit tcp any any eq 11000
ap(config)#access-list 111 permit UDP any any eq 15000
ap(config)#access-list 112 permit UDP any any eq 16000
ap(config)#access-list 113 permit UDP any any eq 20000
```

Figure 5.7: Automatic Traffic Classes Configuration

The Access Point. The Access Point (AP) Cisco Aironet 1131AG is part of the Cisco Aironet series which support IEEE 802.11 a/b/g. Thus, in theory the AP is capable to transmit at 108Mbps. In our scenario, we used this AP to create the wireless network. The access point also supports the Wi-Fi Multimedia (WMM) specification, which is a subset of the IEEE 802.11e standard. As previously mentioned, IEEE 802.11e standard offers mechanism that support QoS requirements of multimedia applications.

To offer QoS at layer 2 (MAC layer), it is necessary to configure the AP in order to allow data frames to be processed according to its traffic class and to certain restrictions defined in access control lists (ACL). The following paragraphs show the ACL configuration in the AP.

Access Control Lists (ACL). ACL allow to filter the network traffic based on IP addresses, network and transport protocols as well transport ports. The commands in Figure 5.7 show how to create ACLs that process network traffic according to the used transport protocol, the IP precedence field, and the destination port.

In our tests, we present 13 ACLs of which 8 ACLs use the IP precedence field and 5 ACLs use the destination port. It is important to remember that the Autonomic QoS Manager invokes this automatic configuration in the UPnP QoS Device (see Figure 4.10).

After defining the ACLs, it is necessary to define the traffic classes that will be differentiated by the AP. Figure 5.8 shows the commands to create the traffic classes and how they are associated to an ACL.

```
ap(config)#class-map match-all best
ap(config-cmap)#match access-group 103
ap(config)#class-map match-all back
ap(config-cmap)#match access-group 101
ap(config)#class-map match-all video
ap(config-cmap)#match access-group 112
ap(config)#class-map match-all voice
ap(config-cmap)#match access-group 107
```

Figure 5.8: Automatic ACL Configuration

Once the traffic classes are created, they are associated to one specific treatment. In order to achieve this, the voice traffic class will be classified with a high priority queue with a Class of Service (CoS) value of 7. The video traffic class is associated to a CoS of 5, thus with a lower priority than voice traffic. The best effort traffic is associated to a CoS of 0 while the background traffic corresponds to a CoS 1. Figure 5.9 shows the command to do the AP configuration. The set of rules is named QoSWMM.

```
ap(config)#policy-map QoSWMM
ap(config-pmap)#class best
ap(config-pmap-c)#set cos 0
ap(config-pmap)#class back
ap(config-pmap-c)#set cos 1
ap(config-pmap)#class video
ap(config-pmap-c)#set cos 5
ap(config-pmap)#class voice
ap(config-pmap-c)#set cos 7
```

Figure 5.9: Automatic Traffic Classes Policy Configuration

Finally, the policies are applied to the AP interfaces in order to treat and filter the network traffic. Figure 5.10 shows the command to apply the policy of traffic treatment to a network interface.

```
ap(config-if)#service-policy output QoSWMM
```

Figure 5.10: Automatic Traffic Classes Policy Assignation

5.3.4 Experiments and Results

This section shows the results of a set of tests intended to validate the autonomic QoS management of a home network based on MODA framework. Figure 5.11 depicts the scenario in which we have developed our tests.

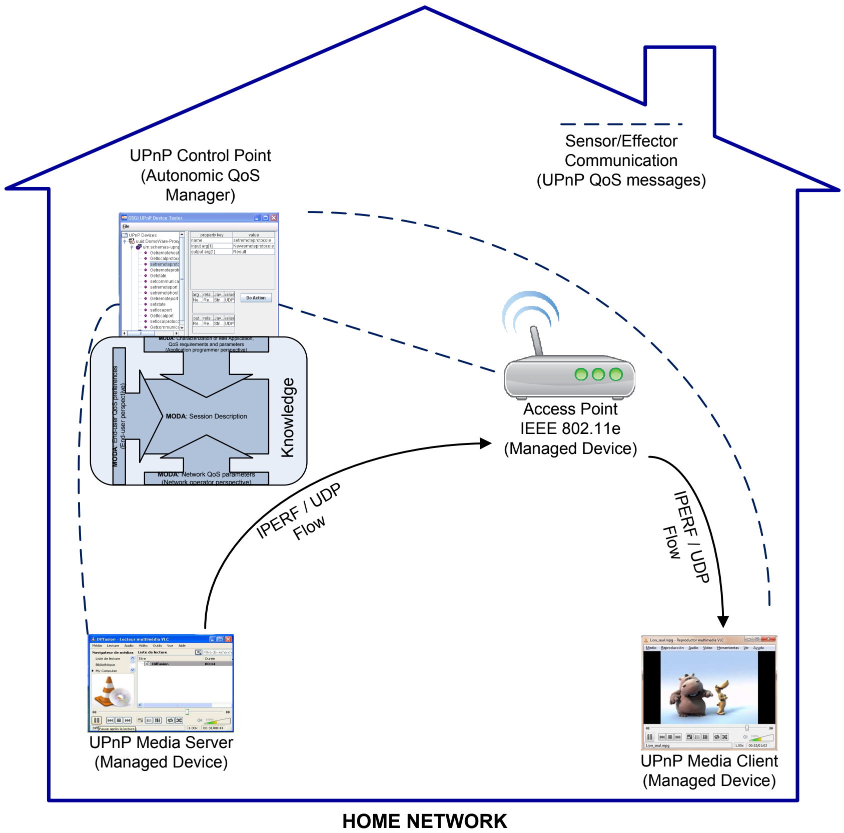

Figure 5.11: Test scenario

Based on the test scenario, experiments have been carried out in an evaluation platform composed by the following elements of hardware and software:

- The Access Point. It is compliant with IEEE 802.11 a/b/g and WMM.

- Control point. It is the UPnP control point running along with MODA and the QoS manager.

- IPERF. It is a testing tool to create traffic in a network like UDP streams and measure network performance.
- VLC. It is a media player. It also performs as a streaming media server. We have used VLC in the conducted subjective test.

Several flows have been generated using the traffic generation tool IPERF[1]. The flows' characteristics are similar to video, voice and streaming applications:

- Streaming: UDP protocol, bandwidth 500kbps, packet size 721 Bytes, TOS: 0x80 => precedence 4 or port dest: 15000
- Video: UDP protocol, bandwidth 1Mbps, packet size 721 Bytes, TOS: 0xA8 => precedence 5 or port dest: 16000
- Voice: UDP protocol, bandwidth 80kbps, packet size of 100 Bytes, TOS: 0xE8 => precedence 7 or port dest: 20000

Streaming with a 20Mbps background traffic				
Slot Time (ms)	**Jitter without QoS (ms)**	**PLR without QoS (%loss)**	**Jitter with QoS (ms)**	**PLR with QoS (%loss)**
0-20	3.097	437/1035 (42%)	0.45	0
20-40	2.906	1538/2434 (63%)	0.331	0
40-60	3.312	742/1730 (43%)	0.574	0
60-80	8.007	612/1768 (35%)	0.416	0
80-100	3.375	776/1702 (46%)	0.497	0
100-120	3.169	720/1732 (42%)	0.65	0
120-140	3.157	733/1733 (42%)	0.413	0
140-160	3.503	742/1737 (43%)	0.239	0

Table 5.2: Results of streaming test with QoS and without QoS

For each one of these flow transmissions, a two phases test has been carried out in order to evaluate the QoS provisioning offered by the QoS manager. For each test, during their first phase, the test has been carried out without any kind of QoS provisioning. In the second phase, the autonomic QoS provisioning is enabled. A 20Mbps background traffic has been produced in order to create network congestion during both phases of each test. We have observed two QoS parameters: the delay variation (jitter) and Packet Loss Ratio (PLR).

Table 5.2 shows the results obtained from the transmission of a streaming flow. From the results, we observe that even if jitter values are below the 10s during both phases of the test (without and with QoS provisioning), we have obtained better

[1]http://sourceforge.net/projects/iperf/

jitter values when QoS provisioning is enabled. In the case of PLR, clearly the percentage of PLR is unacceptable when there is no QoS provisioning.

After testing the QoS provisioning with the transmission of a streaming flow, we have performed another test but now for the transmission of a video flow. Table 5.3 shows the results of this test. As we can see, the 20Mbps background traffic affects the jitter parameter and specially the PLR when QoS provisioning is not enabled during the first phase of the test. When QoS provisioning is enabled, jitter is negligible and an interesting 0% of PLR is obtained.

Video Transmission with a 20Mbps background traffic				
Slot Time (ms)	**Jitter without QoS (ms)**	**PLR without QoS (%loss)**	**Jitter with QoS (ms)**	**PLR with QoS (%loss)**
0-20	3.226	1508/3465 (44%)	0.52	0
20-40	1.904	1445/3468 (42%)	0.756	0.029%
40-60	1.683	1463/3380 (43%)	0.747	0
60-80	1.555	1663/3555 (47%)	0.732	0
80-100	3.156	1260/3464 (36%)	0.786	0
100-120	3.024	1527/3473 (45%)	0.787	0
120-140	2.952	1582/3463 (46%)	0.501	0
140-160	2.895	1541/3467 (44%)	0.619	0

Table 5.3: Results of video transmission test with QoS and without QoS

One more test has been performed for the transmission of a voice flow. During the first phase, the QoS manager has not been activated. The second phase includes QoS provisioning. The 20Mbps background traffic has been maintained in both cases. Table 5.4 shows the results of this test. The results have shown that QoS is enforced by the autonomic QoS manager and the managed elements (QoS devices) in all three tests.

Voice Transmission with a 20Mbps background traffic				
Slot Time (ms)	**Jitter without QoS (ms)**	**PLR without QoS (%loss)**	**Jitter with QoS (ms)**	**PLR with QoS (%loss)**
0-20	2.846	971/2004 (48%)	0.344	0.05%
20-40	2.784	584/1999 (29%)	0.343	0
40-60	3.319	818/2000 (41%)	0.302	0
60-80	3.244	867/2001 (43%)	0.354	0
80-100	4.075	833/1991 (42%)	0.431	0
100-120	3.108	867/2007 (43%)	0.423	0
120-140	2.263	980/2001 (49%)	0.506	0
140-160	3.383	397/2002 (20%)	0.329	0

Table 5.4: Results of voice transmission test with QoS and without QoS

Finally, we have performed a subjective evaluation of the QoS offered by the autonomic QoS manager. A video flow was transmitted from a VLC media server to a media render in the wireless network with a 20Mbps background traffic. Figure 5.12 shows the two snapshots of the reconstructed video at the media client. We can see that the video es evidently degraded when the transmission does not receive QoS provisioning.

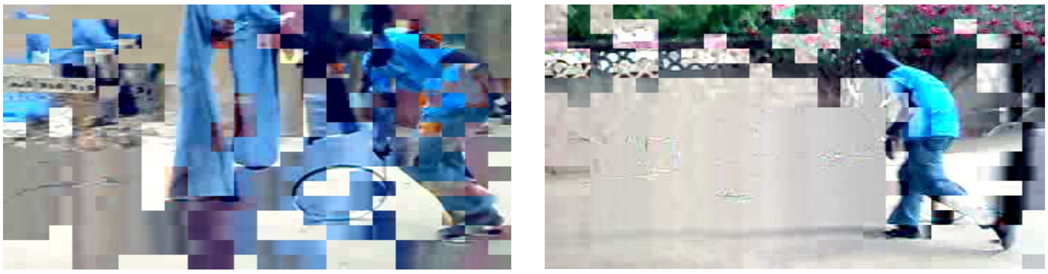

Figure 5.12: Snapshots of the video transmission without QoS (20Mbps background traffic)

Figure 5.13 shows the two snapshots of the same video transmission with QoS provisioning enabled.

Figure 5.13: Snapshots of the video transmission with QoS (20Mbps background traffic)

This scenario of test allowed us to illustrate how the managed devices are self-configured in order to offer QoS mechanisms to applications with specific QoS requirements expressed using MODA.

5.3.5 Summary

This section has presented how our contributions have been deployed and evaluated in the framework of the Feel@Home project. We have presented a simple but common Feel@Home scenario where a user executes Video-on-Demand applications and expresses his/her QoS preferences for both application and user perspectives.

Graph representations of the MODA ontologies have allowed us to illustrate how the semantic representation of the test scenario has been instantiated.

Once the session description can be semantically represented, this information is accessible to the Autonomic QoS framework. As part of the Autonomic QoS Manager, a generic decision model has been defined and implemented. The decision model is part of the Plan function in the MAPE control loop of the Autonomic QoS Manager. In order to show the feasibility of our approach, this generic decision model has been applied to the optimization of network resources utilization taking into account user and traffic priorities.

As a proof of concept, we have presented a case study in which such proposed generic decision model is used. The case study considers user and application priorities, bandwidth constraints according to several multimedia codecs, classes of services, and the decision model. We have used the autonomic QoS manager on a set of simulated flows (with multimedia and non-multimedia characteristics). The results obtained from the tests have allowed us to validate the design and implementation of the autonomic QoS architecture.

This section has also described how to self-configure the UPnP QoS devices (managed elements) based on the information provided by MODA. The technical aspects are presented in order to finally allow the transmission of multimedia flows. We have simulated several kinds of multimedia flows: streaming, video, and voice; all of them with a 20Mbps background traffic. The tests have been carried out with the autonomic QoS manager activated and deactivated in order to be able to observe the behavior of the autonomic QoS manager (Autonomic Manager)and the QoS devices (Managed Elements). Finally, both objective and subjective results were presented.

5.4 Chapter Conclusion

This chapter has presented how the main contribution of this thesis have been deployed and evaluated in the framework of the Feel@Home project. The Feel@Home project has allowed us to implement and to observe our Autonomic QoS provisioning architecture based on the MODA framework.

Here, we remember that autonomic computing architecture is composed by three key entities: the knowledge source, the autonomic manager, and the managed element. We are presenting the conclusion of this chapter from these three perspectives.

From the knowledge source point of view, this chapter has presented how MODA enables the creation of a semantic common space. To do this, we have characterized a Video-on-Demand application from the viewpoints of all the actors in the system. The semantic representations allows us the creation of richer session descriptions which includes, for example, user preferences (QoE), QoS requirements of multimedia applications, communication protocols, codecs, network information, etc. Certainly, MODA is a rich source of knowledge for the autonomic QoS provisioning architecture.

From the point of view of the autonomic manager, the four MAPE operations are considered. Indeed, MODA has a major contribution in the Analysis function of the autonomic QoS manager. We have particularly focused on the Plan function, thus we have proposed a generic decision model implemented in the autonomic QoS manager. This decision model has been used to assign network resources within a simulated congested network. The tests have been conducted for the transmissions of a streaming flow, a video flow, and a voice flow. For each case, the test was developed in two phases: in the first phase, the autonomic QoS manager have not been activated, and in the second phase, it has been activated.

The Monitor and Execution functions are highly related to the Managed Elements (QoS devices) operation. The results assure us to say that the autonomic QoS framework is capable of self-reconfiguring home network devices (Managed Elements) in order to assign shared network resources according to QoS preferences of the users and applications.

CHAPTER 6

Conclusion and Perspectives

Contents

THis thesis has presented an autonomic QoS management architecture based on a semantic source of knowledge called MODA. This research has proposed an autonomic QoS provisioning framework that takes into account the user QoS preferences and application QoS requirements in the context of a home network. The following section presents a summary of the contributions of this thesis, the conclusions as well as several research perspectives.

6.1 Summary of contributions

Nowadays, there is wide deployment of home scenarios where home users execute distributed multimedia applications within their home network. In such contexts, there is a necessity of QoS frameworks that are able to provide the required QoS for multimedia applications while taking into account user and applications QoS preferences and requirements.

Multimedia Ontology Driven Architecture (MODA)

Chapter 3 has presented the current efforts on modeling, programming and deployment of software systems driven by models. Model Driven Architecture (MDA) specifies a methodology for the development of software systems by separating business and application logic from underlying platform technology. MDA starts from high-level representations of a Computing Independent Model (CIM) to Platform Specific Models (PSM) passing through the representation of Platform Independent Models (PIM). Ontology Driven Architecture (ODA) integrates the advantages of Semantic Web technologies with the MDA methodology. ODA proposes the use of ontologies as models to drive the design of architectures, development and deployment of systems. Certainly, the use of ontologies allows the unambiguous representation of domain vocabularies in order to improve the semantics of existing modeling languages like UML. The Ontology Definition Metamodel (ODM) encourages the use of ontologies to drive system architectures. While MDA provides the methodology, ODM provides a family of metamodels, profiles, and mappings among the metamodels for ontology definition. ODM proposes the use of ontologies additionally to UML, in order to have a more complete semantics, then taking advantages of automated inference on models.

Altogether, MDA, ODA, and ODM allow us to define the characteristics that a semantic framework should possess in order to allow the actors (home users, application programmers, and home-network providers) to have a common space, where they can share the semantics of their domain of knowledge. According to ODM's proposition about ontologies applications, our research work corresponds to the area of engineering applications.

Knowledge Representation (KR) is a key component in semantic frameworks. It allows a community of users to have the same understanding about the knowledge being represented. A fundamental part in KR is the utilization of a language to represent the knowledge of a domain. Description Logics (DL) languages have the necessary properties to obtain the required expressiveness when modeling a domain of interest. Additionally, DL languages allow characterizing and storing the knowledge of a domain in a computer readable format. KR also implies organizing the knowledge of a domain. Ontologies help to organize the knowledge but also, they provide the required vocabulary and semantics in knowledge representation.

We have chosen the Web Ontology Language (OWL) as a DL language to develop our ontology-based models. OWL-DL offers a rich expressiveness and reasoning capabilities like subsumption, equivalence, consistency, instantiation checking, and inferences of new conclusions.

Based on these state of the art studies, we have built an ontology-based approach that we have named MODA, the first main contribution of this thesis. MODA is a semantic common space that allows home users and administrators, application programmers, and home-network providers to have a common space, where they can share the semantics about user preferences, multimedia services, and QoS parameters and requirements of multimedia applications. The functionalities of MODA have been presented through the use cases. These use cases are presented from each actor's point of view, which help us to delimitate the specific areas of knowledge that should be characterized. The choice of models included in MODA is driven by the non-functional requirements presented in the use cases. In the context of home networks, MODA is aimed at allowing these actors to share the same meaning of non functional requirements for QoS provisioning.

By using MODA, we have been able to create semantic maps of multimedia sessions in home networks. These maps contain all the necessary information to allow, for example, a home administrator assigning network resources driven by user QoS preferences or by applications QoS requirements. Even, it is possible for an administrator user to manually manage the network resources based on the information provided by MODA, we aimed for the automation of the QoS management in home networks.

Autonomic QoS management architecture

Following the principles of the Autonomic Computing, chapter 4 has introduced our second main contribution: a MODA-based Autonomic QoS Management Framework in the context of home networks. We have proposed an Autonomic QoS management architecture which is able to self-configure the assignation of network resources in order to maintain the required QoS.

In home network contexts, our MODA-based autonomic approach provides the required QoS management by taking into account QoS preferences and requirements from an end-user perspective, multimedia application's QoS requirements, and at the same time, QoS characteristics of the resources offered by the network provider.

We have described how the MODA based autonomic QoS management architecture is used by applying it to the UPnP QoS Architecture. For this purpose, the UPnP QoS architecture has been explained. We have presented a diagram of components in order to show the key elements of the autonomic QoS framework based on MODA and its utilization in conjunction with the UPnP QoS architecture. We have mapped the components of our autonomic QoS provisioning framework to the components of the UPnP QoS architecture. Particularly, we have focused on the Plan operation in the Monitor-Analyze-Plan-Execute (MAPE) control loop of the QoS Manager.

The autonomic QoS management architecture has been presented from four viewpoints. The first one presents the general QoS management architecture from an autonomic point of view. The second point of view presents the mapping between the components of both autonomic computing architecture and UPnP QoS architecture. Such mapping is fundamental in the integration of our autonomic approach with the UPnP QoS architecture. The third point of view corresponds to the Managed Elements (MEs). The MEs are the UPnP QoS devices which are actually the components that enforce the QoS. Finally, the fourth perspective belongs to the actors (end-user, home administrator, application programmer, home network service provider). It presents the process through which the actors interact with MODA.

Deployment and Evaluation

Chapter 5 described how the contributions of this thesis have been deployed and evaluated in the framework of the Feel@Home project. We have presented a simple but common Feel@Home scenario where a user executes Video-on-Demand applications and expresses his/her QoS preferences for both application and user perspectives.

MODA allows creating the session description in a semantic manner. This information is accessible to the Autonomic QoS provisioning framework. As part of the Autonomic QoS Manager, a generic decision model has been defined and implemented. The decision model is part of the Plan function in the MAPE control loop of the Autonomic QoS Manager. In order to show the feasibility of our approach, this generic decision model has been used to self-configure the network resources utilization by taking into account the user and traffic priorities.

As a proof of concept, we have presented a case study in which such proposed generic decision model is used. The case study has considered user and application priorities, bandwidth constraints according to several multimedia codecs, classes of services, and the decision model. The autonomic QoS manager has been used on a set of simulated flows (with multimedia and non-multimedia characteristics). The outcomes from the tests allow us to validate the design and implementation of the autonomic QoS architecture.

From the knowledge source point of view, MODA is a rich source of knowledge for the autonomic QoS management architecture. From the autonomic point of view, the four MAPE operations have been considered. Indeed, MODA has an important contribution in the Analysis function of the autonomic QoS manager. We have particularly focused on the Plan function of MAPE control loop, thus we have proposed a generic decision model implemented in the autonomic QoS manager. The Monitor and Execution functions are highly related to the Managed Elements (QoS devices) operation. The results assure us to say that the autonomic QoS framework is capable of self-reconfiguring home network devices (Managed Elements) in order to assign shared network resources according to QoS preferences of the users and QoS requirements of multimedia applications.

6.2 Perspectives

The outcomes of our research work open important and interesting research perspectives. Here, we present the most significant ones.

Decision models for the Autonomic QoS manager

Decision models integrated in the autonomic QoS provisioning framework could be enhanced in order to integrate more complex utility functions where QoS is provided by computing the *most desired* state from a set of feasible states. Likewise, new decision models could be proposed based on different techniques for example, neural networks whose training could be done using QoE satisfaction. Also, other mathematical/optimization models that take into account concurrent objectives could be integrated, this could allow the autonomic QoS manager to compute a solution in the presence of two or more conflicting objectives.

Integration to End-to-End QoS frameworks

In this thesis, we have addressed the inter-home QoS provisioning. In this context, we have showed that the integration of QoS frameworks that consider QoS preferences of end-users and QoS requirements of multimedia applications is possible. We think that our QoS provisioning approach could be enhanced in order to integrate end-to-end QoS frameworks. Within the framework of the Feel@Home project, another thesis that focuses in Home-to-Home QoS provisioning has been developed. We think that the integration of both intra-home network and inter-home network perspectives represents a complete solution to the QoS provisioning problem considering end-users' QoS preferences and applications' QoS requirements. In this area, the UPnP Forum has recently proposed the UPnP Remote Access Architecture [For11] which aims to allow generic UPnP devices, services and control points to remotely interact with the corresponding UPnP devices, services and control points in a UPnP home network. Here, the extension of our autonomic QoS provisioning approach to inter-home QoS provisioning keeping end-user QoS preferences and applications QoS requirements could be an interesting challenge.

Utilization of MODA in different domains

The design principles of MODA are based on the Model and Ontology Driven Architecture approaches. In addition, the different semantic models of MODA have been constructed based on international standards and recommendations. These two characteristics of MODA facilitate the reutilization of its semantic models in other domains. As an example, in [GMLE09], MODA was re-utilized in order to produce a Service Component Architecture (SCA) deployment descriptor for composing multimedia communication tasks which are defined in the ITU-T Recommendation F.700.

Indeed, MODA can be used to build semantic models aimed at supporting specification and design phases in the software development process, as stated in [Exp10]. In the same work, several semantic models are presented in order to characterize QoS transport mechanisms and services in order to create a composite-based approach for such transport functions e.g., error and congestion control functions. From the application programming point of view, future applications could explicitly integrate MODA-based QoS requirements and constraints. In a similar way, network resources and service providers could provide MODA-based characterization of the network resources they offer. Additionally, more friendly human interfaces could be generated by integrating MODA-base descriptions in its automatic generation. This could facilitate the specification of human-driven policies as well the integration of end-users QoE feedback.

APPENDIX A

Résumé de la thèse

Une Architecture Multimédia Dirigée par les Ontologies pour la Gestion Autonome de la Qualité de Service dans des Réseaux Domestiques.

A.1 Introduction

Les communication réseaux ont vu ces dernières années des changements considérables. A titre d'exemple, il y a un nombre croissant d'utilisateurs sur des nouveaux réseaux qui demandent la possibilité de se connecter à partir d'un large choix de terminaux allant des ordinateurs personnels, aux téléphone mobiles en passant par les ordinateurs portables, les assistants numériques personnels, etc. De plus, on assiste également à un changement dans le besoin des utilisateurs, comme celui de rester connecté à un réseau peu importe le lieu et le moment et le dispositif de communication. Ces nouvelles exigences issues des utilisateurs ont stimulé l'utilisation des réseaux sans fil, mais ont aussi augmenté la complexité de la gestion du réseau et la mise en œuvre de nouveaux protocoles ou services envisageant de nouvelles caractéristiques du réseau telles que la variabilité du trafic.

Aujourd'hui, nous nous rapprochons d'un monde où les éléments matériels et logiciels sont reliés d'une telle manière (filaire, par ondes radio ou infrarouge) qu'ils sont omniprésents et que personne ne s'aperçoit de leur présence. Un monde tel que l'avait décrit Mark Weiser lorsqu'il introduisit sa vision de l'informatique ubiquitaire dans [Wei91].

Dans ce monde d'éléments informatique ubiquitaires et interconnectés, les Systèmes Multimédia Distribués (SMD) sont devenus des systèmes de communication indispensable dans la vie quotidienne des utilisateurs. Par exemple, il est de plus en plus commun d'utiliser des services tels que de la Voix sur IP (VoIP), des appels vidéos provenant de téléphones mobiles (connectés aux réseaux WiFi ou 3G), IPTV, de la vidéo à la demande (VoD), ou les classiques systèmes de visioconférence. En outre, l'utilisation des réseaux sociaux a nettement augmenté l'échange d'information multimédia entre les utilisateurs.

Il existe différents scénarios où l'on peut trouver l'ensemble de ces nouveaux besoins des utilisateurs tous ensembles. Par exemple, c'est le cas d'un utilisateur qui regarde une vidéo *Youtube* depuis sont téléphone portable alors qu'il est en attente à l'aéroport. C'est également le cas des réseaux domestiques lorsqu'un utilisateur joue à des jeux vidéos en réseau depuis son ordinateur tandis qu'un autre membre de sa famille regarde une vidéo à la demande depuis la télévision et qu'une troisième personne participe à une conversation en visioconférence. Ainsi, dans ce genre de contexte, une gestion adéquate des ressources réseaux contribue à fournir une certaine qualité pour le SMD.

La gestion de la QoS pour les applications multimédia avec des contraintes temporelles a toujours été un défi, surtout quand elles s'exécutent sur des réseaux dit de *Best-Effort* [BBC$^+$98, BCS94, RVC01]. Les utilisateurs finaux exigent la même QoS pour leurs applications quel que soit le type de réseau auquel ils sont connectés (WiFi ou 3G) [SABW99, SMH07, HKK04]. En fonction de leurs expériences antérieures, les utilisateurs finaux s'attendent généralement de meilleurs résultats d'un service, par exemple, une meilleure résolution lors d'une visioconférence, ou moins de dégradation sur la qualité d'un streaming audio ou vidéo. Autrement dit, les utilisateurs veulent avoir une meilleure qualité d'Expérience (QoE) [Uni01a, Uni08a]. En effet, la participation des utilisateur finaux à la gestion de la QoS a récemment attiré l'attention de la recherche sur la QoE.

La combinaison du nombre croissant de scénarios ubiquitaires dans lesquels les utilisateurs peuvent s'échanger des informations multimédia, de la complexité de la gestion du système de communication et la prise en compte de la participation des utilisateurs finaux à l'approvisionnement de qualité de service, a exposé de nouveaux défis et créé de nouveaux domaines de recherche qui prétendent à de nouvelles solutions et de nouveaux paradigmes de communication.

Notre travail de recherche est axé sur la gestion de la QoS et de la QoE pour les applications multimédia dans le contexte des réseaux domestiques en tenant compte du points de vue des acteurs impliqués (les utilisateurs finaux, les fournisseurs de réseau, et les programmeurs d'applications). Dans la section suivante, nous présentons plus en détail le contexte de notre travail ainsi que la problématique que nous avons identifiée.

A.2 Contexte et Définition de la Problématique

Tel que nous l'avons mentionné précédemment, ce travail de recherche se situe dans le contexte des réseaux domestiques, dans lequel il est pertinent de se poser la question de la gestion de la QoS. Une question naturelle serait de se demander : *Mais pourquoi?*. D'un point de vue des réseaux domestiques, nous pensons que l'approvisionnement de qualité de service dans Internet est un processus complexe pour les raisons suivantes:

- Il est très difficile d'établir les accords nécessaires entre de nombreux fournisseurs de réseaux. Il y a déjà plusieurs aspects dans lesquels les fournisseurs de réseaux doivent s'entendre avant de déployer une solution de

QoS sur Internet, comme par exemple, les protocoles orientés QoS ou les systèmes de tarification.

- Il y a beaucoup d'utilisateurs qu'il n'est pas évident de différencier lors d'un partage des ressources réseaux.
- Il existe de nombreuses applications et de nouvelles apparaissent chaque jours, ce qui rend très complexe leur traitement vis à vis de leurs exigences.

D'autre part, en ce qui concerne l'approvisionnement de la QoS et la prise en compte de la QoE dans le cadre des réseaux domestiques, nous avons constaté que:

- Il y a généralement un seul fournisseur de service d'Internet (ISP), il est donc plus facile d'établir des accords concernant l'utilisation des ressources réseau.
- Il y a un groupe restreint d'utilisateurs domestiques, ce qui permet l'un d'entre eux de définir une hiérarchie ou d'établir des priorités et jouer en quelques sortes le rôle d'administrateur. Il est également possible pour les utilisateurs d'exprimer leurs préférences ou exigences de QoS/QoE.
- Il y a moins d'applications à fortes contraintes temporelles, il est donc plus facile de les intégrer dans le cadre d'approvisionnement de QoS.

Dans les scénarios de réseaux domestiques, il se peut qu'il y ait plusieurs types d'utilisateurs, par exemple, les enfants, le père, la mère etc. et les utilisateurs peuvent avoir des rôles différents, tels que l'administrateur, l'utilisateur domestique, ou le visiteur. Chaque type d'utilisateur ou chaque rôle d'utilisateur pourrait avoir une priorité différentes et pourraient même assigner des priorités différentes à leurs applications. Dans un réseau domestique, il y a aussi des acteurs autres que les utilisateurs tels que les fournisseurs de services ou les fournisseurs d'accès Internet dont la participation à la gestion de la QoS est indispensable à son existence.

Imaginons maintenant que dans le cas d'un réseau domestique, tous les utilisateurs du réseau domestique soient en train d'exécuter des applications différentes telles que des transferts de fichiers volumineux et des applications multimédias. Supposons qu'à un certain moment, l'administrateur du réseau, constate qu'il souffre d'une certaine dégradation au niveau de la qualité de sa visioconférence. Une solution naturelle serait que l'administrateur du réseau annule toutes les autres sessions (multimédia ou non) existantes dans le réseau afin d'obtenir les ressources réseaux nécessaires pour sa visioconférence prioritaire. Donc l'administrateur pourrait annuler manuellement ou demander à tous les autres utilisateurs d'interrompre leurs applications. Même si cette solution peut être efficace, nous pensons qu'il serait plus judicieux si l'administrateur avait un moyen de prédéfinir le comportement du réseau nécessaire pour maintenir la QoS des applications prioritaires appartenant aux utilisateurs prioritaires.

Nous pouvons voir que dans un tel scénario, l'administrateur du réseau domestique a besoin d'exprimer ses préférences, par exemple, au sujet de l'importance des utilisateurs et/ou des applications. Mais aussi, nous pouvons remarquer qu'il sera nécessaire que le système de communication interprète d'une manière commune avec l'utilisateur, l'expression de ces préférences. Une solution encore plus complète permettrait aux programmeurs d'applications de fixer les exigences de QoS des applications multimédias afin d'assurer la bonne performance des applications critiques.

A.2.1 Semantique

Une définition de la sémantique donnée par le dictionnaire Merriam-Webster[1] est la suivante : "l'étude des significations" ou "la signification des mots". Ainsi, la sémantique implique l'usage d'une langue, de mots, de déclarations, de signes, etc. Dans le cadre des réseaux domestiques, si nous voulons prendre en compte les préférences utilisateurs pour la gestion de la QoS, il est nécessaire que les différents acteurs impliqués aient la même compréhension de ce que les utilisateurs préfèrent, de ce que les applications ont besoin, et de ce que les réseaux peuvent offrir. En d'autres termes, la sémantique joue un rôle important dans notre contexte. Afin de trouver une solution pour ces besoins, nous nous sommes posés les questions suivantes :

- Comment les utilisateurs domestiques peuvent-ils décrire leurs préférences applicatives ?
- Comment les administrateurs de réseaux domestiques peuvent-ils décrire leurs préférences concernant d'autres utilisateurs?
- Comment les programmeurs d'applications peuvent-ils décrire les exigences de QoS des applications multimédias ?
- Comment les ressources disponibles du réseau domestique peuvent être caractérisées ?
- Comment partager une sémantique pour les exigences de QoS des SMD, les préférences de QoE de l'utilisateur, et la caractérisation des ressources du réseau ?

[1] http://www.merriam-webster.com/

Nous avons réalisé qu'il n'y a actuellement pas de solution disponible pour répondre à ces questions. C'est en essayant d'y répondre que nous pouvons nous rendre compte qu'il est nécessaire de décrire ou de caractériser, dans un format lisible par un ordinateur, les applications multimédia ainsi que leurs exigences en terme de qualité de service. En outre, il est nécessaire de modéliser les préférences des utilisateurs et de rendre cette information disponible à une entité gestionnaire de QoS au moment où les ressources réseaux doivent être affectées aux applications. En d'autres termes, nous avons besoin d'un cadre qui est en mesure de représenter les connaissances de chaque acteur et de partager la même sémantique.

Nous pensons que le fait que les utilisateurs, les programmeurs d'applications et les fournisseurs du service de réseaux domestique aient des mécanismes pour décrire et communiquer leurs besoins et les services qu'ils offrent, nous rendra alors capable de profiter de ces informations et de concevoir un cadre générique destiné à gérer la QoS en tenant compte des différents points de vue de ces acteurs. Un tel cadre de QoS permettrait une gestion plus intelligente des ressources du réseau domestique.

Les cadres sémantiques peuvent permettre aux utilisateurs domestiques d'exprimer des exigences ou des préférences de QoE. De même, la sémantique doit permettre aux concepteurs et aux programmeurs de traduire et d'exprimer les besoins et les préférences des utilisateurs en termes d'exigences de QoS. En outre, la sémantique peut aider les systèmes à détecter lorsque ces exigences ou préférences de QoE de l'utilisateur ne sont pas compatibles avec les services offerts par les fournisseurs de services. De la même manière, la sémantique peut aider les concepteurs et les développeurs à concevoir et à développer des applications et des services multimédias en respectant des contraintes de QoS et de QoE. En effet, avoir un moyen pour représenter les préférences et les exigences de QoS dans un format lisible pour les entités informatiques, permettra aux systèmes de les prendre en compte afin de gérer aussi bien les ressources réseau que la QoS.

L'usage des ontologies est une méthode appropriée à l'expression de la sémantique et à la représentation des connaissances d'un domaine [GG95, CJB99]. En effet, les ontologies permettent à une communauté d'utilisateurs de partager la même compréhension de la connaissance qu'elles représentent.

Après avoir réalisé l'état de l'art sur la QoS et la QoE dans le chapitre 2, le chapitre 3 présente les concepts, les définitions, et l'état de l'art de la représentation des connaissances ainsi que la façon dont les ontologies peuvent être utilisées afin d'atteindre les objectifs de cette thèse.

A.2.2 Auto-gestion de la QoS

Revenons à présent au scénario d'un réseau domestique. Nous visons donc une gestion de la QoS dans ce type de réseau en considérant des préférences utilisateurs telles que des priorités entre les applications ou entre les utilisateurs. En effet, en dépit de l'élaboration de protocoles orientés QoS au niveau des couches réseau et transport [BBC+98, BCS94, RVC01, IAC99], l'approvisionnement de la QoS pour ces SMD n'est pas une tâche triviale. Dans un réseau domestique, l'administrateur peut libérer manuellement les ressources réseaux nécessaires en vue de les attribuer aux utilisateurs les plus importants ou aux applications multimédias les plus critiques. De toute évidence, une telle approche n'offre aucune solution performante ni pragmatique.

Même dans le contexte des réseaux domestiques, les préférences utilisateurs peuvent changer de manière dynamique. La priorité d'une application compte tenu des autres applications pouvant s'exécuter en parallèle peut également évoluer. De plus, un utilisateur peut être amené à revoir la priorité d'une application en fonction de son expérience d'utilisation. Ainsi, nous souhaitons proposer une solution qui est capable de gérer la QoS, avec, comme sans l'intervention minimale de l'utilisateur. Autrement dit, nous sommes à la recherche d'un cadre qui est capable de s'adapter dynamiquement afin de gérer la QoS dans un réseau domestique. Plus précisément, nous sommes à la recherche d'un cadre autonome de QoS qui est capable d'adapter l'assignation des ressources réseaux en fonction des préférences et des priorités utilisateurs vis à vis des applications qu'ils utilisent.

En supposant que nous avons déjà résolu le problème de partage des connaissances et de sémantique, nous pouvons nous poser la question suivante : *Comment gérer automatiquement les ressources réseau en fonction des préférences utilisateurs et des exigences de QoS de l'application dans un réseau domestique ?*

Comme une question de fait, cette question nous amène à nous en poser d'autres telles que :

- Quelles informations sont nécessaires afin d'auto-configurer le réseau de communication ?

- Comment auto-configurer le réseau de communication afin de faire respecter les exigences et les préférences de QoS/QoE ?

Le paradigme de l'*Autonomic Computing* [Hor01] pose les auto-propriétés des systèmes autonomes. Il en existe quatre générales, à savoir: l'auto-configuration, l'auto-guérison, l'auto-optimisation et l'auto-protection [IBM03]. Dans un cadre autonome, il est nécessaire de proposer de nouveaux paradigmes de réseaux et communication qui permettent l'auto-caractéristique des architectures de communication et la réduction de la complexité de gestion des réseaux pour offrir une intégration transparente des objectifs des utilisateurs et des contraintes du système. Actuellement, la gestion de la QoS pour les SMD sous une approche autonome reste un problème ouvert. Dans cette thèse, afin de fournir la QoS dans les réseaux domestiques, nous nous intéressons également à l'auto-gestion.

Dans le cadre des problèmes abordés dans cette thèse, nous avons identifié l'élaboration d'un cadre autonome qui est capable d'auto-gérer la QoS pour les SMD dans les réseaux domestiques. Grâce à l'utilisation d'un cadre basé sur les ontologies comme moyen commun de représentation des connaissances et de sémantique (systèmes multimédias, exigences de QoS, préférences utilisateurs etc), nous croyons qu'il sera possible de concevoir des composants de réseaux autonomes qui mettront en œuvre une auto-gestion nécessaire. Le chapitre 4 présente en détail les concepts, les définitions, et l'état de l'art du domaine de l'*Autonomic Computing* ainsi que la façon dont cette approche sera utilisée pour atteindre les objectifs présentés dans la section suivante.

A.3 Objectifs

Pour résoudre les problèmes présentés dans la section précédente, nous précisons ici le but principal de ce travail de recherche.

Nous visons à fournir la QoS pour les systèmes multimédias distribués en développant un cadre basé sur les ontologies qui permette aux utilisateurs finaux et aux développeurs d'applications d'exprimer leurs besoins et préférences de QoS, et de caractériser les services de communication (réseau, transport, middleware) afin d'obtenir la connaissance nécessaire pour fournir la gestion autonome de la QoS dans le contexte des réseaux domestiques.

Pour atteindre l'objectif mentionné ci-dessus, nous proposons les sous-objectifs suivants:

- Développer un cadre basé sur les ontologies qui intègre les différents points de vue des acteurs (e.g. services de communication, utilisateurs, programmeurs d'applications). Avec un tel cadre sémantique, les acteurs seront capables de décrire leurs besoins et de les communiquer entre eux quand il le sera nécessaire. En outre, les services disponibles seront caractérisés, en conséquence, ils seront utilisés à des fins de gestion.

- Développer une architecture pour la gestion autonome de la QoS, qui permettra d'exécuter des décisions précises d'une manière autonome sur la base des informations fournies par le cadre sémantique.

- Évaluer la fonctionnalité du cadre sémantique et la performance du cadre de gestion autonome de la QoS.

Le défi de recherche que nous proposons de résoudre est le suivant:

Un cadre basé sur les ontologies peut permettre aux utilisateurs finaux, aux programmeurs d'applications, et aux fournisseurs de réseaux domestique de décrire *et de communiquer leurs préférences et leurs exigences ainsi que de caractériser les ressources réseaux disponibles. Sur la base de cette sémantique commune, une gestion autonome de la QoS dirigée par les préférences utilisateurs et les exigences d'applications dans le contexte des réseaux domestiques peut être mise en œuvre.*

A.4 Contributions

Comme nous l'avons mentionné, dans le cadre de réseaux domestiques, la participation des utilisateurs finaux à la gestion de la QoS est un défi intéressant. Nous avons décidé de construire une solution afin de résoudre les problèmes identifiés dans cette thèse en deux parties. Tout d'abord, nous pensons que pour tenir compte des préférences utilisateurs et des exigences de QoS des applications multimédias, il est nécessaire de fournir un espace sémantique partagé par tous les acteurs. Deuxièmement, nous pensons que la proposition d'une approche autonome permettra de gérer la QoS dans les réseaux domestiques tout en tenant compte des préférences utilisateurs et des exigences applicatives.

En conséquence, dans cette thèse, nous avons développé les deux contributions principales suivantes :

- Nous proposons une approche basée sur les ontologies nommé MODA, pour *Multimedia Ontology Driven Architecture*. L'objectif principal de MODA est de permettre aux utilisateurs domestiques, aux programmeurs d'applications et aux fournisseurs de réseau domestique d'avoir un espace commun, où ils peuvent partager une sémantique sur les services multimédia, les paramètres de QoS et les exigences de QoS des applications multimédias.

- Nous proposons une architecture pour la gestion autonome de la QoS, qui est capable de prendre en compte les préférences des utilisateurs et les exigences applicatives pour affecter les ressources du réseau domestique. Basé sur la connaissance fournie par MODA, notre approche permet d'intégrer dynamiquement toute l'information sémantique nécessaire à l'auto-gestion des ressources du réseau domestique.

Les contributions mentionnées ci-dessus ont été mises en œuvre et évalués dans le cadre du projet *Feel@Home* [BMMMMD11]. Le chapitre 5 présente le projet en lui-même ainsi que la mise en œuvre et l'évaluation de nos contributions.

A.5 Organisation de la thèse

La structure de cette thèse est la suivante. Le **chapitre 2** présente un état de l'art général des domaines de la QoE et de la QoS. Nous commençons par présenter les définitions de la QoE et leurs importances. Puis, nous présentons les mécanismes existants pour évaluer de manière subjective et objective la QoE. Dans la deuxième partie du chapitre, nous présentons la définition et la synthèse de la QoS. Les cadres de QoS les plus importants, à différents niveaux de la pile réseau, sont brièvement présentés, ainsi que les concepts principaux et les solutions de QoS dans le cadre des réseaux domestiques.

Le **chapitre 3** comporte trois sections principales. La première section présente un état de l'art sur les Architectures et l'Ingénierie Dirigées par des Modèles, plus précisément, sur l'utilisation actuelle de modèles ainsi que sur le support qu'ils fournissent dans le processus de conception, le développement et le déploiement de logiciels. La deuxième section présente l'état de l'art dans le domaine de la représentation des connaissances. Elle se concentre particulièrement sur les ontologies comme un moyen pour représenter la connaissance d'un domaine. De plus, elle décrit le *Web Ontologie Language* (OWL) et ses propriétés en tant que langue de logique de description. La troisième section présente l'une des principales contributions de cette thèse, un cadre basé sur les ontologies que nous avons nommé *Multimedia Ontology Driven Architecture* (MODA). MODA vise à fournir un espace partagé des connaissances entre tous les acteurs dans le contexte des réseaux domestiques.

Le **chapitre 4** contient deux sections principales. La première section présente l'état de l'art sur l'*Autonomic Computing* et l'*Autonomic Communications*. Les bénéfices et les avantages de l'*Autonomic Computing* en ce qui concerne nos problèmes de recherche sont présentés. Dans cette section, on présente l'explication des raisons pour lesquelles le paradigme de l'*Autonomic Computing* s'adapte bien à nos travaux de recherche. La deuxième section présente la deuxième contribution principale de cette thèse: un cadre d'approvisionnement autonome de QoS dans le contexte des réseaux domestiques. Afin de montrer la faisabilité de notre approche autonome, une simulation du gestionnaire autonome QoS est présenté comme une preuve de concept.

Le **chapitre 5** décrit, en deux sections principales, le déploiement et l'évaluation de nos contributions: l'architecture de gestion autonome de la QoS basée sur MODA. La première section présente le projet Feel@Home, qui encadre notre contexte d'étude et contient des scénarios de qualité de service dans lesquels nous sommes intéressés. La deuxième section présente le déploiement et l'évaluation de notre travail. Un étude de cas d'une vidéo à la demande (VoD) est présenté afin de montrer comment instancier les ontologies inclus dans MODA, ainsi que montrer la façon de créer une description complète de la session des applications multimédias, y compris ses aspects non-fonctionnels. Puis, on présente comment fournir d'une manière autonome la QoS dans un scénario de réseau domestique. A la fin de ce chapitre, les résultats obtenus à partir de nos tests sont présentés et analysés.

Enfin, la conclusion générale de cette thèse est présentée ainsi que plusieurs perspectives de recherche.

References

[All07] Digital Living Network Alliance. DLNA Overview and Vision Whitepaper. White paper, DLNA, 2007. Available online (23 pages) http://www.dlna.org/news/DLNA_white_paper.pdf. (Cited on pages 26, 27 and 30.)

[AMA+99] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for Traffic Engineering Over MPLS. RFC 2702 (Informational), September 1999. (Cited on page 22.)

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), December 1998. Updated by RFC 3260. (Cited on pages 2, 6, 12, 21, 34, 130 and 134.)

[BCM+03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. (Cited on pages 48 and 55.)

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), June 1994. (Cited on pages 2, 6, 12, 21, 130 and 134.)

[BFY+00] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine. A Framework for Integrated Services Operation over Diffserv Networks. RFC 2998 (Informational), November 2000. (Cited on page 21.)

[BL84] R. J. Brachman and H. J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th Nat. Conf. on Artificial Intelligence (AAAI-84)*, pages 34–37, 1984. (Cited on page 43.)

[BL85] R. Brachman and H. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1985. (Cited on page 43.)

[BMMMMD11] M. Bel-Martin, G. Maestro-Molina, M. Mahdi, and O. Dugeon. *Digital Home Networking*, chapter The Feel@Home System. ISTE Ltd, Wiley, 2011. (Cited on pages xii, 8, 104, 105 and 136.)

[BVHH+04] S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Stein. OWL Web Ontology Language Reference. W3C Recommendation, 2004. http://www.w3.org/TR/owl-ref/. (Cited on page 45.)

[BZ07] N. Bicocchi and F. Zambonelli. Autonomic communication learns from nature. *Potentials, IEEE*, 26(6):42 –46, nov.-dec. 2007. (Cited on page 81.)

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), September 1997. Updated by RFCs 2750, 3936, 4495, 5946. (Cited on page 21.)

[CJB99] B. Chandrasekaran, J. Josephson, and V. Benjamins. What are ontologies, and why do we need them? *Intelligent Systems and their Applications, IEEE*, 14(1):20–26, jan/feb 1999. (Cited on pages 5, 44 and 133.)

[Com10] International Electrotechnical Commission. Function blocks (FB) for process control - Part 3: Electronic Device Description Language (EDDL). IEC 61804-3, November 2010. (Cited on page 72.)

[CS10] G. Camarillo and H. Schulzrinne. The Session Description Protocol (SDP) Grouping Framework. RFC 5888 (Proposed Standard), June 2010. http://www.ietf.org/rfc/rfc5888.txt. (Cited on page 74.)

[DDF+06] Simon Dobson, Spyros Denazis, Antonio Fernández, Dominique Gaïti, Erol Gelenbe, Fabio Massacci, Paddy Nixon, Fabrice Saffre, Nikita Schmidt, and Franco Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1:223–259, December 2006. (Cited on page 81.)

[DK10] H. Dibowski and K. Kabitzsch. Ontology-based device descriptions and triple store based device repository for automation devices. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1 –9, sept. 2010. (Cited on page 72.)

[DSS93] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation ? *AI Magazine*, 14(1):17–33, 1993. (Cited on page 42.)

[EE09] Jorge Gómez-Montalvo Ernesto Exposito, Myriam Lamolle. Introducing an Ontology Driven Architecture for Distributed Multimedia Systems Engineering. *International Journal of Web Application*, 1(4):228–240, 2009. (Cited on page 95.)

[EGL09] E. Exposito, J. Gomez, and M. Lamolle. Semantic and Architectural Framework for Autonomic Transport Services. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, pages 99–104, nov. 2009. (Cited on page 95.)

[EGM10] E. Exposito and J. Gomez-Montalvo. An Ontology-Based Framework for Autonomous QoS Management in Home Networks. In *Networking and Services (ICNS), 2010 Sixth International Conference on*, pages 117 –122, march 2010. (Cited on pages 87 and 95.)

[Exp10] E. Exposito. Métodologie, modèles et paradigms pour la conception d'une couche transport de nouvelle génération. Habilitation à diriger des recherches, 2010. (Cited on page 128.)

[For08a] UPnP Forum. UPnP AV Architecture for Universal Plug and Play Version 1.0, September 2008. (Cited on pages 28 and 89.)

[For08b] UPnP Forum. UPnP-Device Architeture 1.1, October 2008. (Cited on page 89.)

[For08c] UPnP Forum. UPnP-QoS Architecture:3 For UPnP Version 1.0 Status: Standardized DCP, November 2008. (Cited on pages 12, 26, 27 and 28.)

[For11] UPnP Forum. Remote Access Architecture 2, April 2011. (Cited on page 127.)

[GCTD06] Nico Goeminne, Kristof Cauwel, Filip De Turck, and Bart Dhoedt. Deploying qos sensitive services in osgi enabled home networks based on upnp. In *International Conference on Internet Computing'06*, pages 212–216, 2006. (Cited on page 89.)

[GF92] M. Genesereth and R. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Stanford University, 1992. http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps. (Cited on page 45.)

[GG95] N. Guarino and P. Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. *Towards Very Large Knowledge Bases*, 1(9):25–32, 1995. (Cited on pages 5, 44 and 133.)

[GME10] Jorge Gómez-Montalvo and Ernesto Exposito. A semantic approach to user-based qos provision for multimedia services in home networks. In *Communication Theory, Reliability, and Quality of Service (CTRQ), 2010 Third International Conference on*, pages 150–155, june 2010. (Cited on page 95.)

[GMEL09] J. Gómez-Montalvo, E. Exposito, and M. Lamolle. Towards user-centric configuration and deployment of multimedia services: A semantic framework. In *Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on*, pages 1–7, oct. 2009. (Cited on page 95.)

[GMLE09] J. Gómez-Montalvo, M. Lamolle, and E. Exposito. A Multimedia Ontology Driven Architecture framework (MODA) for networked multimedia systems. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 411–416, july 2009. (Cited on pages 95 and 127.)

[GN87] M. Genesereth and N. Nilsson. *Logical Foundations of Artificial Intelligence*, volume 55. Morgan Kaufmann, 1987. (Cited on page 44.)

[GPFDV96] A. Gómez-Pérez, M. Fernández, and A. De Vicente. *Towards a Method to Conceptualize Domain Ontologies*, pages 41–51. European Coordinating Committee for Artificial Intelligence (ECCAI), 1996. (Cited on page 44.)

[Gru93] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5-6):907–928, 1993. (Cited on page 44.)

[Hay79] P. J. Hayes. The logic of frames. In D. Metzing, editor, *Frame Conceptions and Text Understanding*, page 46. Walter De Gruyer, 1979. (Cited on page 43.)

[HBG+11] D. Hirtle, H. Boley, B. Grosof, M. Kifer, M. Sintek, S. Tabet, G. Wagner, and T. Athan. Schema specification of ruleml 0.91. Specification of The Rule Markup Initiative, May 2011. http://www.ruleml.org/spec. (Cited on page 57.)

[HBL+02] A.P. Hekstra, J.G. Beerends, D. Ledermann, F.E. de Caluwe, S. Kohler, R.H. Koenen, S. Rihs, M. Ehrsam, and D. Schlauss. Pvqm - a perceptual video quality measure. *Signal Processing: Image Communication*, 17(10):781 – 798, 2002. (Cited on page 17.)

[HJP06] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566 (Proposed Standard), jul 2006. http://www.ietf.org/rfc/rfc4566.txt. (Cited on pages 19, 73 and 74.)

[HKK04] Lei Huang, S. Kumar, and C.-C.J. Kuo. Adaptive resource allocation for multimedia qos management in wireless networks. *Vehicular Technology, IEEE Transactions on*, 53(2):547 – 558, march 2004. (Cited on pages 2 and 130.)

[Hor01] P. Horn. Autonomic Computing: IBM's Perspective on the State of Information Technology, October 2001. (Cited on pages 6, 80 and 134.)

[HPSB+04] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Submission, Feb 2004. http://www.w3.org/Submission/SWRL/. (Cited on page 57.)

[HPW00] M. Handley, C. Perkins, and E. Whelan. Session Announcement Protocol. RFC 2974 (Experimental), October 2000. http://www.ietf.org/rfc/rfc2974.txt. (Cited on page 74.)

[HSSR99] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. RFC 2543 (Proposed Standard), Mar 1999. http://www.ietf.org/rfc/rfc2543.txt. (Cited on pages 19 and 74.)

[IAC99] Sami Iren, Paul D. Amer, and Phillip T. Conrad. The transport layer: tutorial and survey. *ACM Comput. Surv.*, 31:360–404, December 1999. (Cited on pages 6, 20 and 134.)

[IBM03] IBM. An architectural blueprint for autonomic computing. white paper, April 2003. (Cited on pages xi, 6, 80, 81, 82 and 134.)

[IEE05] IEEE. Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11

Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 8 Medium Access Control (MAC) Quality of Service Enhancements. IEEE Standard 802.11e-2005, September 2005. (Cited on page 12.)

[IEE06] IEEE. Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks. IEEE Standard 802.1Q-2005, March 2006. (Cited on pages 12 and 23.)

[Ini06] Home Gateway Initiative. Home Gateway Technical Requirements: Release 1, Ver. 1.0. White paper, HGI, July 2006. Available online (112 pages) http://www.homegatewayinitiative.org/publis/HGI_V1.0.pdf. (Cited on page 27.)

[JKK10] Seong-Ho Jeong, Ki Jong Koo, and Do Young Kim. Application level qos control in wireless lans. In *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, pages 210 –213, june 2010. (Cited on page 19.)

[KC03] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, January 2003. (Cited on page 82.)

[KKK+11] Ki Jong Koo, Dong Yuep Ko, Do Young Kim, Byung Sun Lee, and Seong Ho Jeong. A codec-based qos control mechanism for voice over ieee 802.11 wlan. In *Information Networking (ICOIN), 2011 International Conference on*, pages 504 –508, jan. 2011. (Cited on page 19.)

[KW04] J.O. Kephart and W.E. Walsh. An artificial intelligence perspective on autonomic computing policies. In *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on*, pages 3 – 12, june 2004. (Cited on page 85.)

[LPL08] J.-P. Laulajainen, P. Perala, and A. Laikari. Evaluation of upnp qos framework performance in wireless lan. In *Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on*, pages 1–4, april 2008. (Cited on page 89.)

[Met06] Meta object facility (MOF) 2.0 core specification. OMG Specification, January 2006. http://www.omg.org/spec/MOF/2.0/. (Cited on page 35.)

[Min74] Marvin Minsky. A framework for representing knowledge. Technical report, MIT-AI Laboratory, Cambridge, MA, USA, June 1974. http://web.media.mit.edu/ minsky/papers/Frames/frames.html. (Cited on page 43.)

[MMC06] M. Menth, R. Martin, and J. Charzinski. Capacity overprovisioning for networks with resilience requirements. *SIGCOMM Comput. Commun. Rev.*, 36:87–98, August 2006. (Cited on page 12.)

[MPSP09] B. Motik, P. Patel-Schneider, and B. Parsia. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation, October 2009. http://www.w3.org/TR/owl2-syntax/. (Cited on page 46.)

[MVH04] D. McGuinness and F. Van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, February 2004. http://www.w3.org/TR/owl-features/. (Cited on page 45.)

[MW11] Merriam-Webster. Merriam-webster.com, Jul 2011. (Cited on page 44.)

[Nah99] K. Nahrstedt. To overprovision or to share via qos-aware resource management? In *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing*, HPDC '99, pages 35–, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on page 12.)

[NHS05] See Leng Ng, S. Hoh, and D. Singh. Effectiveness of adaptive codec switching voip application over heterogeneous networks. In *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, pages 7 pp. –7, nov. 2005. (Cited on page 19.)

[NM01] N. Noy and D. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05 and SMI-2001-0880, Stanford University, 2001. (Cited on page 44.)

[Obj10] Object Constraint Language (OCL) 2.2. OMG Specification, February 2010. http://www.omg.org/spec/OCL/2.2. (Cited on page 35.)

[OMG03] OMG. The Model-Driven Architecture - Guide Version 1.0.1. OMG Specification, 2003. (Cited on pages 35 and 61.)

[Ont09] Ontology definition metamodel (ODM) 1.0. OMG Specification, May 2009. http://www.omg.org/spec/ODM/1.0/. (Cited on page 37.)

[Pos81] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349. (Cited on page 34.)

[Qui67] M. Ross Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12(5):410–430, 1967. (Cited on page 43.)

[QZ08] Raffaele Quitadamo and Franco Zambonelli. Autonomic communication services: a new challenge for software agents. *Autonomous Agents and Multi-Agent Systems*, 17:457–475, 2008. (Cited on page 81.)

[RNC+09] Stuart J. Russell, Peter Norvig, John F. Candy, Jitendra M. Malik, and Douglas D. Edwards. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., 3rd edition, December 2009. (Cited on pages 42 and 85.)

[RS02] J. Rosenberg and H. Schulzrinne. An Offer/Answer Model with Session Description Protocol (SDP). RFC 3264 (Proposed Standard), June 2002. http://www.ietf.org/rfc/rfc3264.txt. (Cited on page 74.)

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001. Updated by RFC 6178. (Cited on pages 2, 6, 22, 130 and 134.)

[SABW99] S. Sen, A. Arunachalam, K. Basu, and M. Wernik. A qos management framework for 3g wireless networks. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1273 –1277 vol.3, 1999. (Cited on pages 2 and 130.)

[SCFJ03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761, 6051, 6222. (Cited on page 19.)

[SD10] Sanaa Sharafeddine and Zaher Dawy. Robust network dimensioning for realtime services over ip networks with traffic deviation. *Computer Communications*, 33(8):976 – 983, 2010. (Cited on page 12.)

[SMH07] A. Spenst, J. Miroll, and T. Herfet. An implementation of the user-centric qos management approach in wireless home networks. In *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, pages 107 –112, oct. 2007. (Cited on pages 2 and 130.)

[SPTU05] Roy Sterritt, Manish Parashar, Huaglory Tianfield, and Rainer Unland. A concise introduction to autonomic computing. *Adv. Eng. Inform.*, 19:181–187, July 2005. (Cited on page 82.)

[SRL98] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326 (Proposed Standard), April 1998. http://www.ietf.org/rfc/rfc2326.txt. (Cited on page 74.)

[SSS91] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, 1991. (Cited on pages 48 and 56.)

[Ste05] Roy Sterritt. Autonomic computing. *Innovations in Systems and Software Engineering*, 1:79–88, 2005. (Cited on page 82.)

[Tia03] H. Tianfield. Multi-agent autonomic architecture and its application in e-medicine. In *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pages 601 – 604, oct. 2003. (Cited on page 82.)

[TPO+06] P. Tetlow, J. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. W3C Working Draft, 2006. http://www.w3.org/2001/sw/BestPractices/SE/ODA/. (Cited on pages 35 and 61.)

[UG96] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–136, 1996. (Cited on pages 44 and 45.)

[Uni94] International Telecommunication Union. TELEPHONE NETWORK AND ISDN QUALITY OF SERVICE, NETWORK MANAGEMENT AND TRAFFIC ENGINEERING. TERMS AND DEFINITIONS RELATED TO QUALITY OF SERVICE AND NETWORK PERFORMANCE INCLUDING DEPENDABILITY. ITU-T Rec. E.800, August 1994. (Cited on page 18.)

[Uni95] International Telecommunication Union. Data Networks and Open Systems Communications Open Systems Interconnection - Service Definitions Information Technology - Open Systems Interconnection - Transport Service Definition. ITU-T Recommendation X.214, November 1995. (Cited on page 20.)

[Uni96] International Telecommunication Union. SERIES P: TELEPHONE TRANSMISSION QUALITY. Methods for objective and subjective assessment of quality. Methods for subjective determination of transmission quality . ITU-T Recommendation P.800, August 1996. (Cited on page 15.)

[Uni97] International Telecommunication Union. ITU-T Recommendation X.641 Information Technology, Quality of Service Framework, December 1997. (Cited on page 65.)

[Uni00] International Telecommunication Union. ITU-T Recommendation F.700: Framework Recommendation for Multimedia Services, November 2000. (Cited on page 62.)

[Uni01a] International Telecommunication Union. SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS Quality of service and performance. Communications quality of service: A framework and definitions. ITU-T Recommendation G.1000, November 2001. (Cited on pages 2, 13, 14, 15, 18 and 130.)

[Uni01b] International Telecommunication Union. SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS Quality of service and performance. End-user multimedia QoS categories. ITU-T Recommendation G.1010, November 2001. (Cited on pages 23 and 107.)

[Uni06a] International Telecommunication Union. SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS Vocabulary and effects of transmission parameters on customer opinion of transmission quality SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS International telephone connections and circuits - General definitions Vocabulary for performance and quality of service. Recommendation ITU-T P.10 G.100. ITU-T Recommendation P.10 G.100, July 2006. (Cited on page 15.)

[Uni06b] International Telecommunication Union. SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS. Methods for objective and subjective assessment of quality. ITU-T Recommendation P.800.1, July 2006. (Cited on pages 15 and 16.)

[Uni08a] International Telecommunication Union. SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS Vocabulary and effects of transmission parameters on customer opinion of transmission quality SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS International telephone connections and circuits - General definitions Vocabulary for performance and quality of service. Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10 G.100. ITU-T Recommendation P.10 G.100 Amendment 2, July 2008. (Cited on pages 2, 14 and 130.)

[Uni08b] International Telecommunication Union. SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS. Broadband, triple-play and advanced multimedia services - Advanced multimedia services and applications. A generic home network architecture with support for multimedia services. ITU-T Recommendation H.622, June 2008. (Cited on page 27.)

[Uni08c] International Telecommunication Union. SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS. Measurement of the quality of service. Objective perceptual multimedia video quality measurement in the presence of a full reference. ITU-T Recommendation ITU-T J.247, August 2008. (Cited on page 16.)

[Uni08d] International Telecommunication Union. SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS Audiovisual quality in multimedia services Subjective video quality assessment methods for multimedia applications. ITU-T Recommendation P.910, April 2008. (Cited on page 15.)

[Uni09] International Telecommunication Union. SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND

SECURITY. Open distributed processing. Information technology - Open Distributed Processing - Reference model: Foundations. Recommendation ITU-T X.902, October 2009. (Cited on page 18.)

[Uni10] Unified Modeling Language (UML) 2.3. OMG Specification, May 2010. http://www.omg.org/spec/UML/2.3/. (Cited on page 35.)

[VKvBG95] A. Vogel, B. Kerherve, G. von Bochmann, and J. Gecsei. Distributed multimedia and qos: a survey. *Multimedia, IEEE*, 2(2):10–19, summer 1995. (Cited on page 18.)

[Wag08] D. Wagelaar. *Platform Ontologies for the Model-Driven Architecture"*. PhD thesis, Faculty of Science of the Vrije Universiteit Brussel, 2008. (Cited on page 36.)

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, September 1991. (Cited on pages 1 and 129.)

[WHW+04] S.R. White, J.E. Hanson, I. Whalley, D.M. Chess, and J.O. Kephart. An architectural approach to autonomic computing. In *Autonomic Computing, 2004. Proceedings. International Conference on*, pages 2–9, may 2004. (Cited on page 85.)

[ZW00] Schulzrinne H. Zhao W, Olshefski D. Internet Quality of Service: an Overview. Technical Report Technical Report CUCS-003-00, Columbia University, February 2000. (Cited on page 22.)