Semantic Framework for Managing Privacy Policies in Ambient Intelligence

Olfa Mabrouki

▶ **To cite this version:**

Olfa Mabrouki. Semantic Framework for Managing Privacy Policies in Ambient Intelligence. Ubiquitous Computing. Université Paris Sud - Paris XI, 2014. English. NNT : 2014PA112319 . tel-01141997

HAL Id: tel-01141997

https://theses.hal.science/tel-01141997

Submitted on 14 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE : Sciences et Technologie de l'Information, des Télécommunications et des Systèmes

Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi)

DISCIPLINE : Informatique

THÈSE DE DOCTORAT

soutenue le 20/11/2014

par

Olfa MABROUKI

Semantic Framework For Managing Privacy Policies In Ambient Intelligence

Directeur de thèse :	Yacine AMIRAT	Professeur (Université Paris-Est Créteil)
Composition du jury :		
Président du jury :	Nicole LEVY	Professeur (CNAM)
Rapporteurs :	Patrick REIGNIER	Professeur (Université Joseph Fourier)
	Saïd TAZI	Maître de Conférences, HDR (Université Toulouse 1)
Examinateurs :	Yacine BELLIK	Maître de Conférence, HDR (Université Paris-Sud)
	Abdelghani CHIBANI	Maître de Conférence (Université Paris-Est Créteil)

Abstract

This thesis aims at proposing a semantic framework that integrates a meta-model and reasoning tools allowing any ubiquitous system designer to easily implement mechanisms to manage privacy policies. The proposed framework includes a generic middleware architecture that provides components to define, manage and monitor the implementation of privacy policies. Our approach is an hybrid one based on Model-Driven Engineering and a reasoning based on ontologies and inference rules operating on the assumption of the closed world. The proposed meta-model is characterized by a high level of abstraction and expressiveness to define privacy policies management regardless of the domain application and can be adapted to different contexts. It defines, also, a conceptual framework for generic decidable modelling rules to make consistent control decisions on user privacy. These model rules are implemented using the SmartRules language that could implement an adaptive control. The latter is based on a non-monotonic reasoning and representation of instances of concepts according to the unique name assumption. We have validated the proposed semantic framework through a typical scenario that implements support ambient intelligence privacy-aware services for elderly.

Keywords: *privacy, ambient intelligence, MDE/MDA, reasoning.*

Résumé

L'objectif de ce travail de thèse est de proposer un canevas sémantique intégrant un méta-modèle et des outils de raisonnement permettant à tout concepteur de système ubiquitaire de mettre en oeuvre facilement des mécanismes de gestion des politiques de la vie privée. Le canevas proposé intègre une architecture middleware générique qui offre des composants pour définir, administrer et contrôler l'application des politiques de confidentialité. Notre approche proposée est hybride. Elle est fondée sur l'ingénierie dirigée par les modèles et sur un raisonnement à base d'ontologies et de règles d'inférence opérant selon l'hypothèse du monde clos. Le méta-modèle proposé est caractérisé par un niveau d'abstraction et d'expressivité élevé permettant de définir des politiques de gestion de la vie privée indépendamment du domaine d'application pouvant être adaptées à différents contextes. Il définit, aussi, un cadre conceptuel pour établir des modèles de règles génériques et décidables permettant de prendre des décisions de contrôle cohérentes pour la protection de la vie privée. Ces modèles de règles sont mis en oeuvre grâce au langage de règles SmartRules permettant de mettre en oeuvre un contrôle adaptatif. Ce dernier est basé sur un raisonnement non-monotone et une représentation des instances de concepts selon la supposition du nom unique. Nous avons validé le canevas proposé à travers un scénario typique mettant en oeuvre des services d'assistance ambiante sensibles à la vie privée de personne âgée.

Mots clés: *vie privée, intelligence ambiante, MDE/MDA, raisonnement.*

Acknowledgements

I would thank my reviewers, Prof. Nicole LEVY, Prof. Patrick REIGNIER, Prof. Saïd TAZI and Mr Yacine BELLIK for their valuable and insightful comments, which have improved my manuscript substantially.

I am indebted to my supervisor Prof. Yacine Amirat who has been amazingly patient and supportive during my research. He guided me throughout my graduate work, and constantly encouraged me to perform high quality research. I would, also, like to thank Mr. Abdelghani Chibani for his comments and suggestions regarding my work on privacy in Ambient Intelligence. All the thanks to M. Reda BENDRAOU for his help and encouragement and Mrs Véronique VEQUE for her patience and tolerance. Millions of thanks go to Slim SOUISSI for the endless English corrections (thank you very much for reading this manuscript), which taught me a lot and helped improve all my writings.

I would like to thank my mother Zakia, a pillar of strength, for her constant encouragement, her support and her love. Without her, it had not been possible to carry on my doctoral studies. I want to be, also, grateful to my awesome son, Pasha, source of my inspiration and happiness in this world. I want to dedicate this thesis to my sister Ferdaws, my brother Dhafer and for the soul of my father Sadok. My dedication also goes to my uncle Mongi. For all my family MABROUKI and HADJ SASSI, I do dedicate this work.

I avail myself of this opportunity to thank all my friends and colleagues at the LISSI lab for their valuable discussions, support and love. I will never forget you. I would like to thank, also, Stephanie KERROMEN, my manager, at the company "PEAKS" for her support all the time inspite of my difficult context. Finally, I want to thank my friends namely Abida and her husband Mohamed DAHMANE.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The massive use of Ambient Intelligence (AmI) applications that provide users with assistive services in everyday life poses a major technological and societal issue which concerns the protection of users' privacy. To cope with this issue, privacy policies must be set up and constraints of ethical and legal orders must be considered. AmI applications operate, usually, on objects that have an impact on the user privacy such as physical objects namely companions robots, cameras, or any other sensors, or logical objects such as calendar, contact book or personal medical record. The problem of managing privacy must be considered mandatory at the design phase of ambient intelligence services. This is the principle of " Privacy by Design " that aims to provide designers with a methodology and adequate architecture models to make their services responsive to privacy.

Several challenges must be considered in this context. The first one is the architectural style. It concerns the semantic interoperability of privacy management systems. Indeed, for the business information systems, centralized security and privacy management mechanisms are used. Compared to these systems, an ambient intelligence system is characterized by "peer-to-peer" interactions between end users or between end users and services providers both of them using heterogeneous technologies for service delivery. In this architecture, each peer has its own privacy policy management and monitoring mechanisms. In this context, it is difficult to consider a central element in the ambient intelligence environment to ensure interoperability between different policies of these peers. The second challenge concerns the adaptation of the privacy policies preferences contexts of users. They

are constantly moving, changing environment and often interact remotely with other users and use services provided by heterogeneous systems across the web. Therefore, these users should have simple tools to define adaptive policies to their contexts.

Privacy management covers several aspects corresponding to the following normative terms : control, prohibition, authorization and obligation. On the one hand, these terms are associated to privacy control actions such as disclosure, anonymization and obfuscation of personal data.These control actions are executed during privacy sensitive operations such as observation of the AmI environment with sensors, transmission, modification and storage of personal data. On the other hand, these control actions are associated to opening, restriction or disabling of actuators or sensors present in the ambient environment of the user. For example, the access to the camera of the companion robot is disabled when the robot follows the user to the bathroom. The complete functional coverage of all these aspects in the implementation of each ambient intelligence application is an important issue that requires a lot of time and attention in the design time.

The privacy management approaches, proposed in the state of the art, agree on the need of suitable architectures and standardized access control languages for privacy policies management that can be interpreted in the same way by heterogeneous systems. These approaches can be classified into three main categories. The first category concerns the markup languages for defining the consent rules for the access or the use of data exchanged with web servers. The second category concerns the use of language to express XACML policy management of privacy models based on conventional access control RBAC and ABAC. To overcome the problem of semantic interoperability of privacy policies, the third category of approaches aims at providing platforms for semantic representation and reasoning about access control policies expressed in the ontology language OWL/RDF. Although these approaches provide a satisfactory level of expressiveness to define policies, they do not provide a guidance or appropriate engineering tools to facilitate the implementation of the privacy policies management for the corresponding systems. In addition, these approaches are mostly based on an open world assumption which is used with monotonic reasoning. Such reasoning paradigm is useful only for research purposes when we need to query the policy and not to use it as monitoring knowledge to trigger control actions.

To meet all the challenges listed above, we propose in this thesis a semantic framework incorporating a meta-model and a middleware that helps any ubiquitous system designer to easily implement mechanisms to manage users' privacy policies that are effective, adaptive and semantically interoperable. Context awareness is of paramount importance in our approach, through which we aim at exploiting the conceptual description and generic contextual knowledge to develop an adaptative models for privacy policies management. The latter are defined on the conceptual level by using the conceptual entities given within the meta-model or within the derived models. The proposed framework, also, incorporates a generic middleware architecture that provides components to define, manage and monitor the implementation of privacy policies as inference rules. The latter are derived from the policy model in the form of if-then production rules according to the policy authoring templates provided with meta-model. The middleware is also an executable environment that includes two components : a Policy Decision Point (PDP) and Policy Administration Point (PAP). Both of the two components rely on an inference engine that is used to fire the policy rules when facts are asserted in the knowledge base.

To do so, we have adopted an hybrid approach based on Model-Driven Engineering (MDE) approach and a reasoning based on ontologies and inference rules operating on the assumption of the closed world. To overcome the limitations of existing approaches, we have relied on the standard "Model-Driven Architecture" (MDA) of the "Object Management Group" (OMG) to define a meta-model for defining intelligible privacy policies by heterogeneous systems. The proposed meta-model is characterized by a high level of abstraction and expressiveness to define privacy policies management regardless of the domain application and can be adapted to different contexts. It inherits the concepts and rules of two standardized OMG meta-models, namely, the Ontology Definition Meta-model (ODM) and the Business Process Definition Meta-model (BPDM). ODM allows the use all manufacturers of OWL ontologies to represent the UML notation. BPDM can model privacy policies management obligations such as using process control operators.

The Object Constraint Language (OCL) is used to define constraints on the structure of privacy policies or on the architecture of privacy-aware systems. According to MDA, the constraints defined in the meta-model are valid in all models instantiated at the M1 level and implemented in the past from the M0 level systems. The

proposed meta-model, in this case, defines a conceptual framework for modelling generic decidable rules to make consistent control decisions about the privacy. These model rules are implemented using the SmartRules language that has been developed under the SembySem project. This language allows to meet the limitations of using the Semantic Web Rule Language (SWRL). It allows also defining rules based on the negation of facts and allows forward chaining inference using Rete algorithms. These latter are decidable and particularly well suited to handle dynamic knowledge bases and the rules model proposed in our semantic framework. Furthermore, the use of variables in the SmartRules concepts rules can implement adaptive control, which is based on a non-monotonic reasoning and representation of instances of concepts according to the unique name assumption.

We have validated the proposed semantic framework through a typical scenario that implements ambient intelligence privacy-aware services for elderly. The development work concerns, mainly, the emergency case of elderly with the respect to the MDA development process. This latter includes the Computational Independent Model (CIM), the Platform Independent Model (PIM) and Platform Specific Model (PSM).

The thesis is composed of five chapters. This chapter provides a general introduction that introduces the context of our study, the contributions of our thesis and the organization of the dissertation.

In the second chapter, we are going to throw light on the basic concepts related to our thesis. Then, we are going to move on to listing the main applications of the ambient intelligence. Afterwards, we are going to define the social connectedness and social robot in interaction with the elderly as a part of the ambient intelligence. Then, we are going to highlight the different ethical issues and define the privacy management in ambient intelligence environments. Finally, we will end up by casting light on privacy challenges in the ambient intelligence.

In the third chapter, we are going to touch upon the state of the art of the privacy control approaches and privacy-aware systems based on privacy policy languages, access control languages and ontology languages. To begin with, we are going to define privacy requirements, privacy by design approach and privacy modeling techniques. Then, we are going to study the main privacy management techniques according to privacy challenges defined in the previous chapter. Finally, we are

going to classify them through an analysis framework to make the right decision about our privacy contribution in ambient intelligence environments.

In the fourth chapter, we are going to describe our contribution in the field of privacy management in the ambient intelligence. We will present a semantic framework and its integrated meta-model and middleware. We will describe the different layers of this framework and focus on the Model-Driven Engineering, Ontology Definition Meta-model and Business Process Definition Meta-model. Afterwards, we are going to present the semantic executable platform for mapping privacy policies. This platform is based on a reasoning based on ontologies and inference rules operating on the assumption of the closed world using the SmartRules language. In the last part of the chapter, we are going to show how privacy policy templates play a role in the expressiveness of the proposed meta-model.

In the fifth chapter, we are going to implement and deal with the experimental validation of our proposed framework.

At the end, we conclude this thesis with a review of our contributions and we present the main research perspectives that we plan to investigate on the mid term and on the long term.

This work is funded by the *ANRT (National Association for Research and Technology, http://www.anrt.asso.fr)* [1], Grant CIFRE–665/2008 and *CityPassenger*, a french SME. The CIFRE is for *Industrial Conventions for Training through Research*. A Part of the research activity was hold in several european projects at Citypassenger or at the *LISSI (Laboratoire Images, Signaux et Systèmes Intelligents)* laboratory on the topics of context-aware systems and policy based management of privacy and security. The list of the deliverables that are related to this work and the conference papers that are published during the thesis are given hereafter. Two journal papers dealing with the thesis are currently on progress and will be submitted soon.

Olfa Mabrouki, Abdelghani Chibani, Privacy Management in the IoT and M2M Communications, Deliverable 5.5, WP5, M2M security Framework detailed specification, ITEA 2 Project A2Nets, Edited by Hervé Ganem, Gemalto, 2014.

Olfa Mabrouki, Sofiane Bouznad, Abdelghani Chibani, deliverable D4.2, WP5, Study of the appropriate reasoning engines for policy reasoning, ITEA 2 PREDYKOT, Edited by Abdelghani Chiban, LISSI, 2013.

Olfa Mabrouki, Fatma Gharsalli, Walid Tfaili, Abdelghani Chibani, deliverable D5.1, WP5, Multi domain Policies , an Ontology proposal, ITEA 2 Multipl, Edited by Abdelghani Chiban, LISSI, 2009.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, "Privacy in Pervasive Social Networks", Proc. Of AmI 2011 Workshops, Amsterdam, The Netherlands, November 16-18, 2011, Springer Berlin Heidelberg, Intitulé du livre " Constructing Ambient Intelligence", section " Communications in Computer and Information Science ", Volume 277, pp. 296-301, 2012.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, Monica Valenzuela Fernandez, Mariano Navarro de la Cruz, "Context-Aware Collaborative Platform in Rural

[1] *The French ANRT (Association Nationale de la Recherche et de la Technologie.)*

Living Labs", Proc. of the 6th International International Workshop on Cooperation & Interoperability - Architecture & Ontology CIAO! 2010, St. Gallen, Switzerland, June 4-5 2010, Springer Berlin Heidelberg, Intitulé du livre " Advances in Enterprise Engineering IV", Volume 49, 2010, pp 65-76, 2010.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, Monica Valenzuela Fernandez, Mariano Navarro de la Cruz, " Experiments of context-aware collaborative platform in rural living labs", Proc. Of eChallenges 2010, Warsaw, Poland, 27-29 Oct. 2010 , IEEE Computer Society, pp. 1–7.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, Monica Valenzuela Fernandez, Mariano Navarro de la Cruz, " Context-aware Framework for Rural Living Labs", Proc. Of 19th International Conference on Software Engineering and Data Engineering 2010 (SEDE-2010), San Francisco, California, USA, 16-18 June 2010, pp. 146-151.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, Monica Valenzuela Fernandez, Mariano Navarro de la Cruz, "", Ubiquitous Computing and Communication (UBICC) Journal, "Special Issue for Future internet of people", things and services (iopts) eco-systems workshop,15/06/2010. pp.1-8.

Olfa Mabrouki, Abdelghani Chibani, Yacine Amirat, Monica Valenzuela Fernandez, Mariano Navarro de la Cruz, " Context awareness in collaborative platforms and living labs", Proc. of the 3rd International CompanionAble Workshop IoPTS, Brussels, 2nd December 2009, pp19-26.

Chapter 2

Privacy in Ambient Intelligence : Basic Concepts, Technologies and Challenges

Contents

2.1 Introduction

In this chapter, we are going to shed light on the basic concepts related to our thesis starting with the ubiquitous and pervasive computing concepts, moving to pointing out their relationship with the ambient intelligence research area. Then, we are going to move on to listing the main applications of the ambient intelligence, focusing, mainly, on the smart home, domestic care of the elderly, assisted living and healthcare. Then, we are going to define the social connectedness as a part of the ambient intelligence. So, the focus is going to be on the social robot in interaction with the elderly. This leads us to highlight the different ethical issues regarding this kind of application. Another area of study we are going to embark on is the privacy management in ambient intelligence environments especially considering privacy in social interactions and in mobile cloud computing. Finally, we will end up by casting light on privacy challenges in the ambient intelligence.

2.2 Ubiquitous and Pervasive Computing

Ubiquitous Computing (UbiComp) as envisioned by Weiser [1] is a computing environment that integrates technologies in all types of appliances and devices in our daily life (see Figure 2.1). It aims at making computing and communication essentially transparent to users. Indeed, invisibility is the most important aspect of UbiComp. The user is exposed to a few sets of services available to him/her and is oblivious to the complex system implementing those services. Today, UbiComp includes not only an unlimited number of mobile and distributed applications but also the use of the advances of Pervasive Computing (PerCom) to present a global computing environment [2].

Basically, the terms UbiComp and Pervasive computing are used interchangeably [3], but they are conceptually different [4]. Pervasive computing aims at creating

a smart environment with embedded and networked computing devices [5]. It provides human users with seamless service access. Autonomous detection of application requirements and automatic service provisioning are considered as the key features of pervasive computing middleware [5]. In other words, when acquiring context from the environment, PerCom dynamically builds computing models depending on context. Consequently, PerCom aims, also, at creating ambient intelligence where networked devices embedded in the environment provide unobtrusive, continual, and reliable connectivity and also perform value-added services [6].

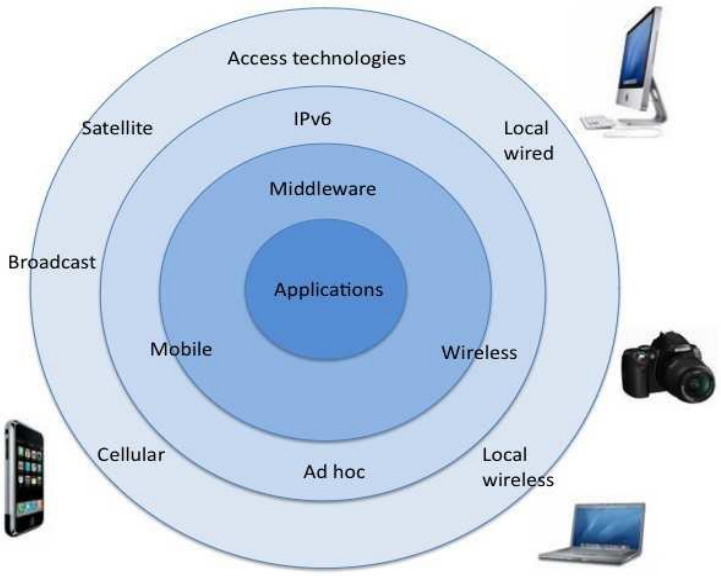

FIGURE 2.1: A General View of Pervasive Computing.

According to the authors in [7], [6], [2], [4],[3],[8],[5], the most important and common features of UbiComp and PerCom are : connectivity everywhere and every time, interoperability, adaptability, proactivity, natural interaction via simple interfaces, context-awareness and privacy-awareness. An important feature of pervasive systems is also the ease of use. Ubiquitous computing has been designed in order to be deployed anywhere and to be accessed by the majority of users (often non-specialists). Used devices and embedded software must be manageable and accessible for all. Consequently, users are connected everywhere and every time to PerCom systems thanks to the mobile devices. Those systems are autonomous enough to capture and analyse contextual parameters of users any time and everywhere. Interoperability is considered if the environment is heterogeneous. Ubiquitous environments include a wide variety of technologies. Thus, the difficulty

lies on creating an intermediate system that works like a gateway, providing opportunities to all heterogeneous physical and software equipments, to interconnect, communicate and interact. Adaptability is considered when PerCom environment provides a personalized interface according to the user context. This requires an adjustment of services and an availability of resources that correspond to user context features such as its location, current activity, preferences, etc.

To better meet users' expectations, and even anticipate their needs proactiveness should be an essential characteristic of systems, software, and devices. This can happen at various levels including the application level, between applications, the operating system level, or the network level. A proactive middleware for pervasive computing manages communication between devices connected to the real world, anticipates user needs, and initiates actions on the user's behalf transparently, assisting the user without distracting him from the task at hand. Thus, the deployment of the technology of natural interaction interfaces should allow handling natural and intuitive : "if it is not easy to use, it will not be used". However, simplicity can be expected to present a major challenge especially when it comes to maintain a balance between accessibility and security.

Moreover, the main feature of UbiComp and PerCom paradigms is the context awareness. Information on the context and situation makes the environment more sensitive. Systems must be able to recognize the context and customize their services according to the user context. It is, therefore, fundamental for ubiquitous technologies to incorporate data acquisition and reasoning tools about the context. In many researches [9], [10], [11], the context is limited to person's location and time. However, there were many efforts to determine the real meaning of context and consequently of context-aware systems. Dey, in [12], considered that "context of an entity is its physical, social, emotional, and mental (focus-of-attention) environments, location and orientation, date and time of day, other objects in the environment". This generic definition of context can be used in any application scenario to specify and enumerate the context's characteristics. Hence, the majority of these researches share a common vision of context as it represents a set of information about location, time and activity of a person.

Otherwise, PerCom systems collect many sensitive data about human users such as their personal information and medical records. Hence, users should be aware of how their information are used or disclosed. Many privacy-aware mechanisms

control data through security policies. Although, enforcing privacy for information requests allows users controlling their information. This approach creates an imbalance between the information that users are willing to share and the data collectors' requirements for service delivery.

2.3 Ambient Intelligence (AmI)

The term "Ambient Intelligence (AmI)" was defined by the European Commission in 2001. In the beginning, Advisory Groups (one of the European Commission framework programme) launched the AmI challenge [13]. Then, the European Community Information Society Technology (ISTAG) updated it in 2003 [14]. Currently, AmI is the intersection of two research areas : ubiquitous and pervasive computing. AmI has, also, a strong relationship with artificial intelligence, wireless networking, human-computer interaction and robotics domains as depicted in the figure 2.2. AmI objective is to make human daily life better by making people's surroundings flexible and adaptive [15]. The recent definition of AmI states that it is a digital environment that proactively, but sensibly, supports people in their daily lives [16] in a non intrusive way [17].

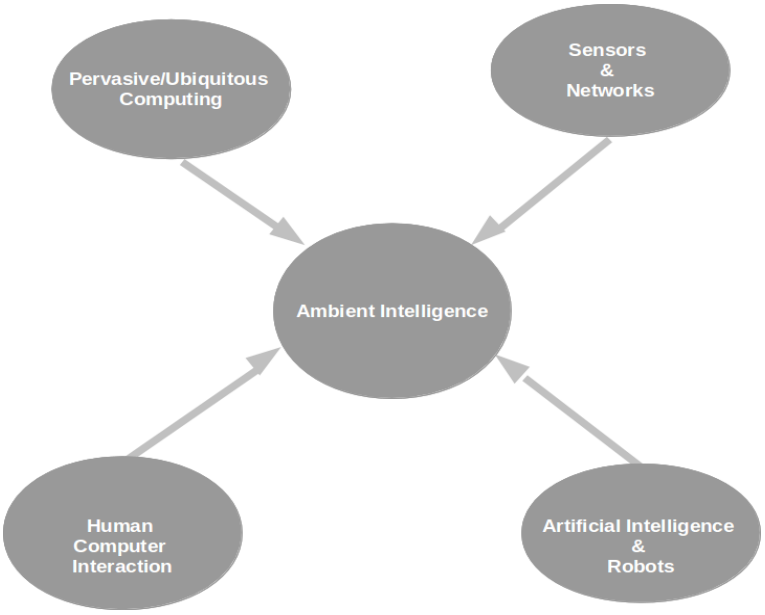

FIGURE 2.2: AmI Research Areas.

In [18], Aarts and Ruyter, considered AmI as an electronic system that is sensitive and responsive to the presence of users. The authors identify five key features

of AmI technologies [19]: embedded, context-aware, personalized, adaptive, and anticipatory. So as far as the definition of embedding is concerned, Aarts et al. mean possibly small miniaturized devices that merge into the background of people activities and environments [18]. Then, AmI system can improve the user's productivity. To do so, the user should be able to communicate with the system through an enhanced user-interface, such as voice recognition.

According to the author, in [20], the most important features are intelligence and embedding. Intelligence allows us to point out that the system is sensitive to the gathered context of user from sensors and network devices. This system is able to be adaptive and learns from the behavior of users, and eventually, recognizes and expresses emotion [20]. Consequently, intelligence is related to the context/situation awareness and the personalization. In addition, AmI features will automate many aspects of our daily life, will increase the productivity at work and even will customize our shopping experiences. There are many applications fields where AmI can be valuable for users. The following is a presentation of some of these applications coupled with a focus on the smart homes and assisted living AmI applications.

- **Transportation and automotive.** Cook et al. argue that transport means are valuable settings for AmI technologies [20]. Public transport can benefit from AmI technology including GPS-based spatial location, vehicle identification and image processing to make transport more fluent and hence more efficient and safe [15].

- **Education.** AmI can help to improve the learning experience for the students. Education-related institutions may use technology to track students progression on their tasks, frequency of attendance to specific places and health related issues like advising on their diet regarding their habits and the class of intakes they opted for [15].

- **Healthcare.** AmI technologies can bring additional benefits by integrating wearable devices such as micro-sensors to the patients and providing further support within normal daily life. These technologies aim at making it easier for individuals to monitor and maintain their own health while enjoying lives in normal social settings.

- **Ambient Assisted Living and Well-being.** Support for independent living for the elderly or in other words assisted living is one of the major application areas of ambient intelligence. This is due to growth of longevity of the populations.

- **Emergency management.** Safety-related services like fire brigades can improve the reaction to a hazard by locating the place more efficiently and also by preparing the way to reach the place in connection with street services. The prison service can also quickly locate a place where a hazard is occurring or is likely to occur and prepare better access to it for personnel security [15].

- **Production-oriented places.** Production-centred places like factories can self-organize according to the production/demand ratio of the goods produced. This will require careful correlation between the collection of data through sensors within the different sections of the production line and the pool of demands via a diagnostic system which can advise the people in charge of the system at a decision-making level [15].

- **Shops, shopping, recommender systems.** Sadri, in [20], looks at shops as responsive environments, with devices controlled by software agents that react to the presence of customers according to the customers' identities and profiles. The idea is to propose an ubiquitous commerce, which brings e-commerce and AmI together, with a framework for context-dependent interaction between shops and shoppers.

2.4 Ambient Intelligence Main Applications

2.4.1 Smart Home

A smart home is one of the famous applications of AmI. Various names have been used to describe homes equipped with pervasive technology to provide AmI services to the inhabitants. Smart homes may be the most popular term, and other terms include aware houses, intelligent homes, integrated environments, alive, interactive, responsive homes/environments. Innovation in domestic technology has long been driven and marketed by the desire to reduce labour and improve the quality of time

spent at home. This continues to be one of the motivations for the development of AmI at home. Other factors include technological advances and expectations, and an increasing trend in a way of life that blurs the boundaries between home, work, rest areas and entertainment.

In [20], Sadri defines smart home as a home equipped with sensors and actuators of various types to monitor activities and movement, and to monitor risk situations, such as fire and smoke alarms. He specifies that the sensors and actuators control household appliances (e.g. cooker and fridge), household goods (e.g., taps, bed and sofa) and temperature handling devices (e.g. air conditioning and radiators) [15]. Lê et al. [21] state that the concept of "smart home" is a subject to various definitions and interpretations because being "smart" can imply various characteristics of a highly advanced modern home, such as being automatic, compact, innovative, convenient, self-adjusting, responsive, or functional. These authors propose a conceptual framework for smart homes characterized by having five basic features as seen in the figure 2.3 : (1) Automation which represents the ability to accommodate automatic devices or perform automatic functions; (2) Multi-functionality that shows the ability to perform various duties or generate various outcomes; (3) Adaptability that aims to make the system able to customize (or to be customized) to meet the needs of users; (4) Interactivity that means the ability to interact with or allow for interaction among users and (5) Efficiency that illustrates the ability to perform functions in a time-saving, cost-saving and convenient manner.

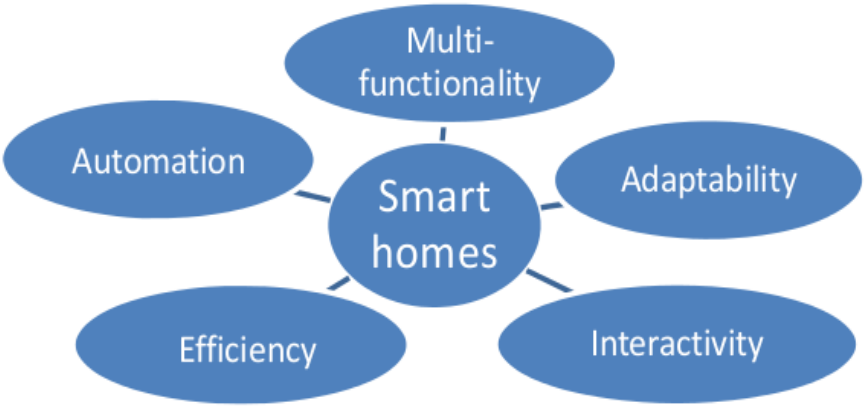

FIGURE 2.3: Conceptual Framework of Smart Homes.

2.4.2 Domestic care of the elderly, Assisted Living and Healthcare

According to the "French National Institute of Statistics and Economic Studies (INSEE) [22]", France population will reach 73.6 million in 2060, approximatively, 11.8 million more than today. The proportion of people aged 60 or over will increase, from 21.7 % reaching 31% in 2035. After 2035, it will still continue to grow. People aged 75 and more represented 5.2 million of the France population in 2007 (8.9% of the population) and will be 11.9 million in 2060 representing (16.2% of the population) and the number of those who are aged 85 and over will increase from 1.3 to 5.4 million, four times more than today. This trend is due to the increase in life expectancy at birth. in 2008 the life expectancy was 77.6 years among men, 84.4 years among women and in 2060 respectively 86 and 91.1 years [23].

Consequently, many eHealth applications have been developed to improve the quality of health care. For example, doctors can view radiological films and pathology slides in remote sites, and assist or perform surgeries via remote robots [24]. Hospitals have increased the efficiency of their services by monitoring patients' health and have progressed by performing automatic analysis of activities in their rooms. They can, also, increase safety by authorizing medical staffs and patients to have access to specific areas and devices [17]. In addition, there is a broad range of available sensor technologies to measure various respiratory, biochemical (e.g. glucose levels), and physiological (e.g. ECG) parameters. Gouaux et al. [25] report the development of an intelligent wearable personal ECG monitor (PEM) as a part of the European EPI-MEDICS project. The need for technology in this area is obvious from looking at our current and project future demographics.

2.5 Social Connectedness

Social connectedness aims at protecting a person against dementia [26]. It improves the pain of cognitive decline for both elders and their caregivers. As a matter of fact, socially isolated elders are vulnerable regarding their emotional and physical health. In addition to its intrinsic value, socializing is a strong motivation for participation in other healthy behaviours. Regarding the nature of social connectedness, most

elders want to feel that they are having an impact on others rather seeing themselves as passive recipients of help as it is stated in [27].

Social interactions through web applications using ubiquitous computing technologies are changing our social practices in the private, friends and professional spheres. Several social interaction platforms and socialwares (e.g., Facebook, MSN, Skype, Youtube, MySpace, Twitter, Google+, etc.) have emerged quickly all over the world to become the main social communication channel. Social wares are various and social practices, build around them, differ a lot depending on the actual application in use and the group using them. Social wares include, naming a few, instant messaging, broadcasting and video casts, micro blogging, web forums and message board, weblogs, wikis, and photo/video sharing. In these platforms, persons can interact together with their contacts (relatives, friends, colleagues, etc.) to share opinions, insights, experiences, and perspectives. Through these interactions, users aim at being a part of the network connections or to be part of a group of interests. Hence, social networks services allow persons to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view their list of connections and those made by others within the system.

The nature and nomenclature of these connections may vary from site to site [28]. Social networks platforms share, approximately, the same principles of displaying friends' lists. These lists contain links to each friend's profiles, enabling viewers to traverse the network graph by clicking through the friends' lists. On most social networks platforms, the friends' list are visible to anyone permitted to view the profile. People can leave messages on their friends' profiles. In addition, social networks often have a private messaging feature similar to webmail. While both private messages and comments are popular on most of the major social networks, they are not universally available [29].

The natural evolution of social networks is envisioned through a full integration with ubiquitous and pervasive computing. Actually, pervasive social network is not just another buzzword but is a well established paradigm. It aims at complementing traditional social networks with services allowing the interactions with real world objects present in the surrounding [30]. In specific terms, persons are using mobile devices (smart phones, smart watches, cameras, etc.) to interact and share their experiences together through the services provided by the AmI environment (e.g.,

display screens, multimedia servers, experience sharing web platforms such as picassa, youtube, etc.) [31]. In pervasive social networks, social sensing refers to interaction between people and objects tracking people. Embedded sensor devices are typically of everyday life devices and appliances such as refrigerators, consumer products, televisions, cars, etc. They may be highly connected and may be used for making smarter and automated decisions [32]. Sensing objects could be mobile phones, wearable sensors and pedometers. Such paradigms have tremendous value in enabling social networking paradigms in conjunction with sensing. The increasing ability of commodity hardware to track a wide variety of real life information such as location, speed, acceleration, sound, video and audio leads to unprecedented opportunity in enabling an increasingly connected and mobile world of users that are ubiquitously connected to the internet [32].

Several approaches were proposed in the state of the art to link social networks with AmI environments. For example, in [30], Ben Mokhtar et al. proposed a social networking middleware service. This latter, dynamically, combines both social and physical proximity relations between mobile users to accurately recommend them to people with whom they perform activities of common interest. The service is based on a social network propagation component that infers users' relations both within the same (intra) and across (inter) users' activities. Ben Mokhtar et al., in [30], evaluated, also, the impact of various middleware deployment strategies on the ability of the social network propagation component to find related users, and analyse the advantages and shortcomings of each of them.

With respect to social networks modelling, individuals are considered as nodes while relationships are considered as "social ties" [33]. Once all nodes and ties are completed, social network can be drawn. The resulting graph-based structures are often very complex. There can be many kinds of ties between the nodes. Several researches have shown that social networks operate on many levels, from families up to the level of nations. They play a critical role in determining how problems are solved, organizations are run, and the degree to which individuals succeed in achieving their goals. Hence, the position for every person in social network is well defined. "Distance" between two people is also known as the "geodesic distance" or "degree of separation". This distance represents the shortest path in the network from one person to another. For example, a person is one degree removed from his/her friend, two degrees removed from his/her friend's friend,

three degrees removed from his/her friend's friend's friend, etc. Social networks contain, also, collections of subnetworks or "components" [33]. A component is a part of a network in which everyone is connected by at least one tie to every other person in the same component. In other words, for two different components, no one in the first component can be connected to anyone in the second component.

AmI technology should aspire to catalyse rather than replace human interactions. The aim is to help people sharing information with others in their social network. Many mechanisms should be involved to fulfil this goal. However, many ethical issues arise regarding users privacy. In the following part, we are going to cast light on the concept of "Social Robotics" as an example of social connectedness and then, point out the consequent ethical issues.

2.5.1 Social Robotics : Social Interactions with and through Ubirobots

Social robotics is a growing field concerned with the issue of how humans and robots can better live, work, and interact together. It involves problems of human perception, human behaviour modelling, action planning in the presence of humans, or design of socially acceptable human-robot-interfaces. Methods from robotics may be combined with models and insights from social psychology and cognitive science. These social robots will be considered as companions at home as they can take care of the elderly or they can, even, help/assist in the service industry. This next generation of robots also involves intelligent transportation systems. Their tasks require advanced social and cognitive skills to effectively interact and cooperate with humans [34]. Hence, many research questions arise about social compatibility, learning human social behaviour and recognizing intentions of the robots as stated in [34].

Social Robotics share the same objectives as ubiquitous robotics research [20]. Ubirobots are cognitive entities able to move around, sense, reason and proactively execute tasks and adapt themselves to the situation they may face anywhere and anytime. Any software agent running on daily living objects such as Smart phone, TV, Oven, Bed, or Office can, also, be considered as an UbiRobots. Hence, these ones are not only limited to physical mobile robots. Chibani et al. consider [20]

that UbiRobots have the particularity to overcome the limitations of stand-alone companion and wearable robots. They well integrate the ambient intelligence and use the web services technologies. Consequently, Social Robot is an Ubiquitous Robot or UbiRobot that has a social interaction with people thanks to the natural language.

Several prototypes of social robots have been built over the world. For example, Keepon is a social robot with the appearance of a tennis ball. It was developed in the National Institute for Interconnecting Technology in collaboration with Carnegie Mellon Institute [18]. Their plan was to build a social robot that could interact with children particularly those with behavioural disorders such as autism. The robot interacts with the child through his sense of rhythm : it moves thanks to the music, the voice, and is able to express emotions such as surprise, envy or joy. The Keepon can, also, respond to children's touch gestures (hugs, pokes, pats and tickles) with emotions and sounds, thus it facilitates its interaction with the child. Paro is a therapeutic robotic companion. It was developed in Intelligent Research Institute [19]. This robot is designed for the elderly, especially, those with the alzheimer disease. These people are often deprived of social ties (in the hospital or at home). The robot aim is to give few smiles to its companion. It responds to touch by movements and cries. Actually, Pero is housed in a French hospital. The iCat is a technology developed by Philips to study human-robot interaction (HRI)[13]. It was sold to more than 40 universities. Among its uses, we can find an iCat playing chess with a child (cooperation between the child and the robot is then studied) or a iCat encouraging a child during a game of chess.

With respect to assistive robotics research, social robot is used in hospitals, galleries, museums, airports or domestic (private) spaces such as offices or homes. Feil-seifer and Matarić, in [14], give a definition of Socially Assistive Robotics (SAR). They describe them as a class of robots that is in the intersection of Assistive Robotics (robots that provide assistance to a user) and Socially Interactive Robotics (SIR robots that communicate with a user through social and non-physical interaction).

The SIR aim to address critical areas and gaps in care by automating supervision, coaching, motivation, and companionship aspects of one-on-one interactions with individuals from various large and growing populations. These latter include stroke survivors, the elderly and individuals with dementia and children with autism spectrum disorders. In this way, robot cists aim is to improve the standard of care

for large user categories. The term of SIR was, also, introduced to distinguish social interaction from teleoperation in Human-Robot Interaction (HRI). Several concerns regarding human perception of robotics, particularly the difference in social sophistication between humans and social robots, were addressed.

2.5.2 Ethical Issues in Ambient Intelligence

ICT (Information and Communication Technologies) change, continuously, the lives of institutions, companies, societies and even of people. Their social impact is particularly significant on application models. If these technologies give several opportunities to explore, evaluate and structure, they also generate new fears, give rise to old fears and can cause further damages. The rise of uncertainty about their applications leads to an increase in security requirements, an obligation information, a right to a "consent". The paradoxical increase in risk leads to a dialogue between the scientific community and citizens. The upheaval is even more important as the complexity of issues to be resolved can lead to the search for self and the production of "private standards" (standard contracts, charters, codes of conduct, etc.).

As stated by Brey and Philip, in [16], one of the fundamental ethical questions regarding AmI is whether it is more likely to enhance human autonomy and freedom, or decrease it. The autonomy called, also, the self-governance is considered as the ability to construct one's own goals and values. It is to have the freedom to make one's own decisions and perform actions based on these decisions. If some people are not autonomous, they are not able to express their preferences and their requirements. Otherwise, the autonomy is strongly related to the freedom. As Isaiah Berlin has argued in a famous essay, freedom comes in two sorts, positive and negative [15]. Negative freedom is the ability to act without obstruction or interference by others. Positive freedom is the ability to be one's own master, having one's own thoughts and making one's own decisions. Negative freedom means that no one stands in your way. Positive freedom means that no one tells you what to think.

Both types of freedom involve control. Positive freedom involves control over the environment. Negative freedom involves self-control, or control over one's own thoughts and decisions. Normally, AmI guarantees for users more control over the environments with which they interact as it will become more responsive to their

needs and intentions. Moreover, these environments may contain, particularly, hundreds of networked computing devices. In such landscape, computers should not wait for human inputs in order to take an action or communicate with another device, but pro-actively anticipate the user's needs and take action on his/ her behalf.

It is mandatory to consider ethical, political and economic choices that engage our societies. The sphere of civil liberties may be, by its direct relation to ICT, a privileged field of investigation. How to preserve the fundamental rights of individuals in their living, especially their dignity, freedom and protection? How to "help", technologically, patients with debilitating diseases without "violating" their right to privacy? Production of rules may come from the legislator, judge and tools of auto-adaptation of AmI applications. Many questions arise regarding the relationship "public space-private space," the concept of finality, the contours of privacy, the biological privacy in however going beyond the simple legal analysis focus on ethical grounds. The aim is not to disturb innovation but to make choices about common values such as integrity of the person, security and trust.

2.6 Privacy Management in AmI Environments

Ambient intelligence involves extensive and invisible integration of computer technologies in people's everyday lives. Such integration will certainly open up issues of privacy, risk, acceptance, identity, trust and security [35]. These key issues have been identified from their earliest inception. The first definition of privacy goes back to the 19th century with Warren and Bradeis who defined privacy as "the right to be let alone" [36]. They have, also, linked privacy to the property that may be tangible or intangible. They acknowledged that the disclosure of personal information is a violation and breach of privacy. Since their definition, many new definitions have been proposed. The one presented by the British Committee on privacy and related material covers the known aspects of privacy [37] : "the right of every individual to be protected against any intrusion into his personal life, career or his family by physical means or via a direct disclosure of information. In [38], privacy is defined as fundamental human right, enshrined in the United Nations Universal Declaration of Human Rights and the European Convention on

Human Rights. Control of information about users is, also, a form of privacy. A taxonomy of privacy focuses on the harms that arise from privacy violations. This can provide a helpful basis on which to develop a risk/benefit analysis.

Evolutions of technology and social changes have, also, affected the privacy of individuals as they provided new ways of intrusions. It is ,then, worth highlighting the aspects of protecting information collected from individuals, their use, and their processing. Westin gave more details on the definition of privacy [39]. He stated that privacy is the claim of individuals, groups and institutions to determine when, how and to what extent information is communicated to others. This is the type of privacy that is particularly relevant in the field of AmI. Sen, in [40], defined privacy as "the ability of individuals to control the way in which their personal information is acquired and used". Westin et al. in [39], defined privacy as the control over the disclosure of information. A more detailed discussion of various aspects of privacy can be found in [40], [41], [42], [43], [44]. Thus, the questions that may arise are : (1) How to allow users to control their visibility in AmI spaces as they normally do in physical spaces? (2) How to manage shared information so that boundaries between public and private are blurred? (3) How to convince the users that AmI environment protects adequately their privacy and how a person can ensure the confidentiality of the collected data?

While little privacy architectures have been designed, several principles regarding privacy have been agreed upon [45] : (1) *Notice* concerns the data collection that could be more and more unobtrusive. Sensors can be practically invisible, storing users' information transparently. Notice aim is to notify AmI users when, how and what personal data are gathered. It is commonly agreed that no data may be collected without explicitly stating the collection; (2) *Choice and consent* state that a user must not only be informed about data collection, but also be offered a choice whether or not to use a data-collecting service. If the individual chooses to use the service, this consent should be given explicitly. Marc Langheinrich [46] noticed that requiring consent without providing other options is not a realistic choice. For example, the user can refuse to be tracked by a video camera; (3) *Access* indicates that users must be able to view their data, and, also, to correct or delete them. Basically, users should be in control of their own data; (4) *Anonymity and pseudonymity* is regarding privacy legislation. This latter, usually, lays no restriction on the collection of data if it is not linkable to an individual.

In [47], a set of generic privacy concerns in AmI arise. A pervasive network of interconnected devices and communications will mean that the sheer quantity of personal information in circulation will increase greatly. The introduction of perceptual and biometric interfaces for certain applications will transform the qualitative nature of personal information in circulation. In order to offer personalised services, AmI will require the tracking and collection of significant portions of users everyday activities. Langheinrich has proposed a set of principles and guidelines for the design of privacy-aware pervasive applications much closer to the process of software design in the real world and therefore easier to follow [48]. Hong et al. have proposed a complementary approach based on risk analysis to identify the privacy risks of a pervasive application and how to manage these risks in the application design [49]. Based on the works above, Chung et al. have created a set of design patterns for the development of pervasive or ubiquitous applications, including specific patterns to handle privacy [50].

The presence of sensors and actuators in AmI environments makes them smarter, especially, with important capabilities of information processing. These devices gather human user sensitive information. However, this ease of access to user data threatens, seriously, their privacy that can be exploited by malicious or even by the curious system administrators. Therefore, AmI systems could play the role of a"big brother". Their main role is to capture the maximum of information about users and track them. For example, in the case of hospitals, clinics and homes where there is an abundance of sensitive personal information that must be protected and secured. Authors, in [51], claimed that privacy protection can be divided into several categories : (1) protecting a person's identity; (2) protecting personal data; (3) protecting the actions of an identity and (4) protecting the instructions or tasks of an identity. Authors added that an adequate solution must be found for each of these categories to provide full privacy.

Identity concept is a key element of the definition of user privacy. Identity can be seen from many perspectives : philosophical, psychological, sociological, legal and technical perspectives [51]. We are interested in the technical point of view of identity which concerns user information. Technical identity defines how and by whom that information can be accessed and modified. Otherwise, identity and trust are related concepts. The issue of trust from the users perspective deserves greater consideration as AmI users can consider it as a base for its privacy. User is

aware of his/her privacy and believes that it has to be respected. Building trusty capabilities and detailed study will emphasize the trust in information systems [51]. Therefore, the risk of identity theft can be diminished to gain user's trust.

2.6.1 Privacy in Social Interactions

Privacy concerns appeared in the popular press. In [52], Alison was wondering about sharing personal information with total strangers on the internet in social networks. He stated the dangers regarding having public user profile, especially, children who are well publicized. He added that the sheer volume of personal information that people are publishing online is changing the nature of personal privacy. It is easy to differentiate the professional, personal and family life. However, with the spread of social networks everywhere on internet, it is difficult to make a difference between all of these aspects. Hence, the distinctions become blurred. Alison concluded his article by claiming that "Anything you put on the internet has the potential to be made public". In another paper, in the USA Today [53], Kornblum and B. Marklein highlighted privacy problem in socialwares. Users of social networks assume that when they share their photos on web sites, only their friends could see them. However, it is not the case. Additionally, they proclaimed that for teens and young adults, socialwares are private spaces where they can interact in the open, multimedia style of the online world in which they grew up. However, for adults, these sites are places where kids are putting their reputations and future at risk.

Many researchers have handled the potential threats to privacy associated with social networks. In one of the academic studies on privacy on Facebook [54], Gross and Acquisti studied patterns of information revelation in online social networks and their privacy implications. In addition, they analyzed the online behaviour of more than 4,000 Carnegie Mellon University students who have joined Facebook. They evaluated the amount of information they disclosed and studied their usage of the site's privacy settings. Moreover, they highlighted potential attacks on various aspects of their privacy, and showed that only a minimal percentage of users changed the highly permeable privacy preferences. A lot of attention has been paid to Facebook as the number of its members has exceeded the one million in the recent years. This socialware presents, in addition, several privacy problems. According

to Acquisti and Gross [55], Facebook offers attractive means for interaction and communication. However, it raises privacy and security concerns. They made a survey for Facebook members at a US academic institution. They analysed the impact of privacy concerns on members' behaviours. Then, they compared members' stated attitudes with actual behaviour. As a result of their study, they found that an individual's privacy concerns are only a weak predictor of his membership to the network. They also found evidence of members' misconceptions about the online community's actual size and composition, and about the visibility of members' profiles.

When analyzing trust on social networks., Dwyer et al. have conducted an online survey of Facebook and Twitter [56]. In their study, they compared perceptions of trust and privacy concerns, along with willingness to share information and develop new relationships in these socialwares. As a result of their work, members of both socialwares reported similar levels of privacy concern. Their members significantly expressed greater trust in both Facebook and Twitter. These results showed that, in online interaction, building new relationships means automatically trust for social networks users. In another study [57], Jagatice et al. pointed out the issue of "phishing" because of accessible profiles on social networks. They accused social networks of being an easy way to improve the effectiveness of attacks by a quantifiable amount. Furthermore, privacy is involved in users' ability to manage their social contexts. Preibusch et al. argued that the privacy options offered by social networks do not provide users with the flexibility they need to handle conflicts with friends who have different conceptions of privacy [58]. Hence, they proposed analysis framework for privacy in social networks as a solution to these conflicts.

2.6.2 Privacy in Mobile Cloud Computing

Mobile cloud computing refers to the availability of cloud computing services in a mobile environment [38]. Commonly, this means that an application can run on a remote resource rich server like Facebooks location aware services, Twitter for mobile, mobile weather widgets etc. Mobile cloud computing incorporates the elements of mobile networks and cloud computing, thereby providing optimal services for mobile users. In mobile cloud computing, mobile devices do not need a

powerful configuration (e.g., CPU speed and memory capacity) since all the data and complicated computing modules can be processed in the clouds. Storing a personal data on a mobile device where they could be accessed by the cloud is one of the key concerns for people about using a mobile cloud. The mobile device can reveal many personal information on people as it contains their contact lists, text messages, personal photos and videos, calendars, location information. Moreover, the cloud services are stated to be vulnerable.

Users may lose their data if the services go out of business, or simply if the services fail due to technological problems. Indeed, users do not own or operate their own data. This introduces privacy issues and can limit users control. In addition to an authorization scheme, users of the mobile cloud should also have the ability to change their privacy settings and state what information can be seen. For example, a mobile cloud participant may not want other devices to record his/her location information [38].

Maintaining the levels of data protection and privacy required by current legislation in cloud computing infrastructure is a new challenge, as it is meeting the restrictions on cross-border data transfer [59]. Privacy issues are central to user concerns about adoption of cloud computing, and unless technological mechanisms to users concerns are introduced. This may enable trust issues to many different types of cloud services. Users fears of leakage of commercially sensitive data and loss of data privacy may be justified : in 2007, the cloud service provider Salesforce.com sent a letter to a million subscribers describing how customer emails and addresses had been stolen by cybercriminals.

2.7 Privacy Challenges in AmI Environments

AmI environments are endowing with sensory functionalities that are able of collecting large amounts of data about people living conditions, their interactions and their everyday activities [35]. Giving people an efficient way of keeping control and managing their privacy is an important requirement that poses several challenges. On the one hand, people expectations and concerns about privacy vary widely and depend on the applications they are using [60] and, on the other hand, AmI environments are composed of heterogeneous systems and evolve continuously.

In this context, it is difficult to find a good balance between the use of traditional access control approach of handling privacy by using priory defined rules and an approach that is user centric, context-aware and adaptive. The latter is too difficult to implement without a strong commitment for application designers and the users themselves. In this section, we are going to highlight the most important challenges of privacy management in AmI.

- **Semantic Policies.** Policy based management of privacy is the natural approach for handling privacy of personal resources. The privacy policy should be defined by the owner of personal resources or the designer of the application. It must control operations on resources on both the owner side or the third party side. The latter is authorized by the owner to access or handle the private resource. Privacy policies must be defined using a high-level language and according to privacy management models. The language must offer meaningful terms and a clear syntax format with formalized semantic in order to be readable by both end users and machines. The models must be independent from the implementation platform as much as possible to allow for a similar enforcement of the policy on both the owner and the third party sides.

  Ontologies, and in particular those defined with semantic web languages, are considered as the most promising framework for addressing such a challenge. A semantic web ontology has several benefits. It provides a common vocabulary with formal semantics that can be used for abstracting the real world heterogeneity. It is used also for expressing semantic knowledge as high-level and well structured statements. The ontology is by nature extensible and has executable platform for making semantic reasoning. These benefits make it possible for the definition of semantic policies and applying an interoperable and adaptive control on the owner and third party sides.

- **Multi-domain Interoperability.** As a matter of fact, the interoperability of privacy control is the most critical challenge that should be solved in order to enable user privacy in AmI environments. AmI services interoperate through web services that are supplied provided by different organizations such as hospitals, security companies, etc. We call these organizations privacy domains and the interoperation is multi-domain. The privacy rules of the

user should comply with the privacy policies of these organizations. The management of privacy policies should be interoperable regarding the domains in order to preserve the anonymity of private context information and identity attributes without decreasing the quality of the provided service.

The specification of privacy rules should provide tools to create a multi-domain privacy policy that can take into account a full, partial and unknown description of local domain policies. The policies that are partially described or unknown for non-disclosure reasons, but are specified and enforced by external systems such as black boxes, should be queried at access control run-time.

- **Fine Grained Privacy Control.** AmI applications have to provide privacy policies tailored to individuals privacy needs. Classic privacy languages or based access control policies do not provide sufficiently fine-grained protection. Non-expert users require a fine-grained access control over their private information because of the heterogeneity and variety of their personal information (e.g., contexts, profiles, photos, and microblogs). These users could specify, effectively, who should have access to which part of their data. Thus, a practical solution requires rather a fine-grained per-data than a per-interface access control. For example, when sharing his current location or activity, the user could choose a sublist of his contact list or make new different lists. Consequently, the users have fine-grained control over any user who can see the divulged information.

- **Management of Obligations.** Obligations are actions that a user has to perform in the system according to the privacy policy. They are, usually, executed when a set of conditions are true. A system should only allow obligations to be assigned when the obligated user will have sufficient privileges in the system and access to the resources necessary to successfully fulfill the obligation. In addition, privacy policy has to care about sequencing of privacy rules or obligations management : what is the priority of a given privacy rule? What is the first privacy rule that has to be executed and what are the next ones? In the classic vision of privacy systems, rules are defined independently from each others. However, privacy control mechanisms require reasoning about a process, a series of temporally constrained actions and occurrences.

- **Context Awareness.** Context-aware systems must be able to apply an adaptive control that depends on owner or third party context. Context awareness is a reasoning process that is used in adapting the enforcement of privacy policy. The context concerns for example location, current activity, situation, vital signals, health records, etc.). Such are processed independently from the application core functions and are used by the application anywhere at any time. A change in context will represent a change in the risk on privacy and the system may adapt the control by disclosing certain information or denying the access to previously authorized access when the risk is high.

  The storage of context history is an important issue. AmI system should, also, keep track and store the history of user privacy preferences and contextual information in the system to avoid setting a privacy policy already defined, avoid conflicts, etc.

- **Adaptability.** In AmI environment, privacy policies should be tailored to users' privacy needs. They should be flexible and consider the change of user privacy preferences. Even, they are defined at design time, privacy management system should be dynamic enough to provide the users with the best way to control their private information. Having static privacy policies that don't consider the previous defined policies and cannot reason about them is a barrier to make the system adaptable. Dynamic separation of duty permits more flexibility in operations. Therefore, we argue that using meta-data can be useful in expressing conflicts semantics. These meta-data should be defined in a separate representation that includes constraints about the access context.

- **Conflicts Management.** User privacy rules should not have mutual conflicts. In practice, this assumption is generally satisfied when the privileges assigned in a given rule are typically disjoint from the ones assigned in another system. Multiple types of conflicts may arise when composing privacy rules. For instance, in structural conflict, positive authorization rule is defined, specifying permitted actions for a target service, whereas the second rule defines a negative authorization rule, specifying forbidden actions on the same target service. In addition, there may be two rules that forbid and permit the same action on the same local resource; or two rules that permit

the same user to have two roles that cannot be assigned together during the same session.

2.8 Summary

In this chapter, we have presented the background of Ubiquitous/Pervasive Computing and Ambient Intelligence (AmI). We have given an overview of the main applications with a particular focus on those having an impact on privacy such as health-care and well being remote monitoring, social connectedness, companion robots and mobile cloud computing for the elderly in particular. As a next step, we moved on to explaining the ethical issues of ubiquitous computing as well as the main concepts and challenges of managing privacy in ambient intelligence. In the next chapter, we are going to explain, in detail, the main requirements of privacy management and present a study of the main approaches for managing user privacy that are proposed in the state of the art.

Chapter 3

Privacy Management in Ambient Intelligence : State of the Art

Contents

3.1 Introduction

In this chapter, we are going to touch upon a state of the art of the privacy control approaches and privacy-aware systems based on privacy policy languages, access control languages and ontology languages. To begin with, we are going to define privacy requirements, privacy by design approach and privacy modeling techniques. Then, we are going to study the main privacy management techniques according to privacy challenges defined in the previous chapter. Finally, we are going to classify them through an analysis framework to make the right decision about our privacy contribution in AmI environments.

3.2 Privacy Requirements Analysis

The design of privacy-aware systems requires the analysis of the privacy according to multiple aspects that depend on the privacy requirements and needs of users and stakeholders. In general, the stakeholders aim is to define privacy conformance system behaviour. They evaluate systems to detect privacy leakages and to calculate

metric values. These results of computation serve as indicators for describing the privacy risk of using such systems, and for checking if a behaviour conforms to a given set of privacy requirements (as verifying system conformance). Hence, privacy requirements are considered very important as well as functional and nonfunctional application requirements such as quality of service, performance and security.

High level privacy requirements have to take into account ethical principles, legal issues and technical application issues. They are defined in several forms : formal privacy criteria of logical constraints or technical policy statements. Thus, privacy analysis calculates the privacy risk and detects privacy breaches of a system/application. It may, also, adapt mechanisms of security and risk analysis research fields. As stated in [61], risk analysis uses and adapts mechanisms of Requirement Engineering, and Secure and Dependable Engineering. In [61], the authors propose a goal-oriented approach for analysing risks during the requirements analysis phase. They analyse risks along with stakeholder interests, and afterwards identify countermeasures and introduce them as part of system's requirements. The work of [61] relates and uses existing work of Requirement Engineering, and Secure and Dependable Engineering.

According to the Privacy Impact Assessment (PIA) [62], the privacy analysis process aim is to create privacy-aware design specifications. The PIA process takes the perspective of legal authorities and project managers. The results of PIA are privacy requirements that consider privacy regulations, privacy laws, and project requirements adapted to the application domain. These results are described at a conceptual level which does not reflect technical interdependencies. Formal methods for privacy analysis are mostly based on languages which describe technical systems. The purpose is to provide a formal defined vocabulary to avoid ambiguity, to make domain assumptions explicit, to prove properties of a solution, and to explore the design space. Different languages (mostly logic based) and existing mechanisms (e.g. model checking or system simulations) which can be used to provide (semi) automatic evaluation and verification of systems. Technical privacy requirements involve system-specific properties but often fail to integrate high-level privacy requirements that include privacy regulations or stakeholders' interests. Further, systems are often too complex to verify their conformance to given constraints on a detailed level.

3.3 Privacy by Design

Privacy by design aims at ensuring privacy and personal control over one's information and at gaining a sustainable competitive advantage for organizations. Privacy should be considered in the early stage of application design. It has to be embedded into the design and architecture of IT systems and ambient intelligence design practices. Design of the privacy follows several principles as stated in [63]. The design of privacy should be proactive. It has to anticipate and prevent privacy invasive events before they happen. It aims to prevent privacy risks and infractions from occurring. Additionally, at this stage, system should offer a minimum of default privacy rules. These ones must deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given system. If a user does nothing, his/her privacy still remains intact. Hence, the system manages users' privacy by the default.

In addition, privacy by design requires keeping the interests of the users uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user friendly options. In October 2010, the International Conference of Data Protection and Privacy Commissioners, which included over 600 representatives of governments, companies, and nongovernmental organizations, approved a resolution which recognizes the concept of "Privacy by Design". The goal is to ensure that privacy is embedded into new technologies and business practices from the outset as an essential component of privacy protection [63]. Privacy protection actions should remain invisible and transparent to users.

3.4 Privacy Modelling

Privacy requires the involvement of various aspects throughout the system life cycle. For instance, skills could concern specifications related to security, to the hardware platform development or to the maintenance of access rights. During the application life cycle, the developers will analyse requirements, develop, and test the system. Security has to be considered from the beginning of the application specification states. Privacy control systems engineering for AmI systems is a complex process. Model Driven Engineering (MDE) techniques are used to handle

such complexity by creating models for abstracting from details, separating logical units (modularization and encapsulation), and documenting the design. During the development of privacy-aware applications and systems, it is necessary to identify and specify required privacy properties from the design stage. Then, these privacy properties or privacy requirements have to be addressed in the implementation phase.

Otherwise, in order to identify and specify privacy criteria, one may use several mechanisms provided by Requirements Engineering (RE) [64]. During the requirement engineering phase, we apply additional technologies as best practices, design pattern, standard models, meta-models, and ontologies to reuse existing solutions, to adapt general (i.e. domain independent) solutions and to provide a conceptualization. It is mandatory to determine privacy control mechanisms, privacy measures and audit regarding the privacy requirements. Bridging the gap between high level requirements and technical issues is, also, indispensable. Hence, privacy-aware application should sustainably detect privacy leakages. Privacy policies are, highly, used to enable user privacy control. An appropriate privacy policy has to match with user privacy requirements and needs. Moreover, semantic technologies, background knowledge, or data mining techniques are used for privacy attacks. Privacy policies should be expressive enough to avoid such attacks.

3.4.1 Semantic Modelling

A variety of terms and concepts must be managed in complex applications. Different approaches to deal with such complexity exist. For instance, we formulate precise and unambiguous description based on logic. We abstract and filter out details to focus on one aspect as a specific problem we want to solve. We present the same content in different ways. A comprehensive approach has to consider the need for a clear conceptualization of the domains and their assumptions. We use ontologies to explicitly describe the conceptualization of a domain. Ontologies are a formal specification about how to represent objects, concepts and other entities that are assumed to exist in an area of interest, as well as the relationships among them. They have a common understandable domain, explicit semantics, expressiveness and enable the sharing of information. According to Gruber [65], the ontology of a shared domain can be described by defining a set of representational terms. These

terms (lexical references) are associated with entities (non lexical referents) in the universe of discourse.

Formal axioms are also introduced to constrain their interpretation and well-formed use. In this respect, ontology is viewed as the explicit statement of a logical theory. Indeed, in the context of the Semantic Web, "ontologies describe domain theories with the intent of the explicit representation of the semantics of the domain data". Although such ontologies often assume a form of a taxonomic class hierarchy, they are by no means not restricted to hierarchies. Indeed, ontologies may take on the form of much more general and complex structures.

Using ontology enables the definition of concepts and relations representing knowledge about a particular document in typical domain terms. In order to express the contents of a document explicitly, it is necessary to create links (associations) between the document and relevant parts of a domain model, i.e. links to those elements of the domain model, which are relevant to the contents of the document. Model elements can also be used for search and retrieval of relevant documents. In case when all documents are linked to the same domain model, it is possible to calculate a similarity between documents using the conceptual structure of this domain model. Such approach supports, also, 'soft' techniques where a search engine can use the domain model to find concepts related to those specified by user.

3.4.2 Model-Driven Engineering

Model-Driven Engineering (MDE) is a software development methodology. It is based on creating and exploiting domain models rather than on the computing concepts. MDE approach provides several benefits [22]: (1) **Abstraction.** The model-driven approach provides an abstract view of the system becoming more and more complex. User can highlight relevant elements in a model or possibly ignore unnecessary details. (2) **Traceability.** Traceability is required between process phases, design artifacts and implementation artifacts. All stakeholders can use a same and unique model but with adapted views throughout the life cycle of AmI application. (3) **Consistency.** During the entire life cycle, consistency is ensured by using a common model which can be checked by verification tools.

(4) **Process guidance.** It is possible to provide context sensitive help during the entire life cycle.

Moreover, code generation and model transformation correspond to powerful tools which limit the design fault risk. In the following, we focus on model-based solutions with security features. In [23], the authors specify a meta-model for a unified access control mechanism. Indeed, access control could be done through several standard mechanisms like RBAC (Role Based Access Control) [66], [67], ORBAC (ORganization-Based Access Control) [68], or XACML (eXtensible Access Control Markup Language) [69]. The model presented is generic and could be mapped into all of these existing models. Then, it is possible to declare software components and specify access control policies. This approach is independent of the platform (PIM – Platform Independent Model) and, during the code generation, three targets (PSM – Platform Specific Model) are proposed.

The Secure UML profile [70] is another profile for access control. All concepts manipulated in the profile are first defined in a meta-model. This profile is based on the RBAC model which is hampered by the non-support of system condition rule definition. This is resolved thanks to the definition of OCL (Object Constraint Language) [71] constraints. Models can be used for specifying threats, vulnerabilities, and security risks. Unified Modelling Language (UML) profile for security assessment, called Security Assessment UML, is proposed in [72]. Compared to the model-based framework proposed by the European IST CORAS project [73], this profile is provided by guidelines. Finally, the UMLSec profile [74] is considered as a main contribution for model-based security. UMLSec is an extension to UML and supports security as a non-functional property. Through several UML diagrams like use case, class and interaction diagrams, it is possible to specify security requirements.

3.4.3 Model-Driven Architecture

In order to use any model with Model-Driven Architecture (MDA)-based tools, it is first necessary to understand each of its technology and standard. In this part, we provide an overview of MDA basics especially : UML, Meta-Object Facility (MOF) and Profiles.

MDA initiative is an approach proposed by the Object Management Group (OMG) [75], [76] in 2001 to system-specification and interoperability based on the use of formal models [77],[78] and [79]. Basically, MDA uses modelling languages to specify a system at several levels : a Computation Independent Model (CIM) to represent the system's environment and requirements, a Platform Independent Model (PIM) that describes the system architecture in a technology-neutral manner, and a Platform Specific Model (PSM) that expands the PIM with details specifying how the model is to be implemented using a specific platform-a set of subsystems and technologies. MDA standards separate business and application logic form underlying platform technology.

PIMs of an application or integrated system's business functionality and behaviour, built using UML and the other associated OMG modeling standards, can be realized through the MDA on virtually any platform, open or proprietary, including Web Services, .NET. CORBA, J2EE, and others as shown in the figure 3.1. These PIMs document the business functionality and behaviour of an application separate from the technology-specific code that implements it, insulating the core of the application from technology [77]. As mentioned in the OMG MDA specification [77], MDA provides an approach and the necessary tools for : (i) specifying a system independently of the platform that supports it; (ii) specifying platforms; (iii) choosing a particular platform for the system and (iv) transforming the system specification into one for a particular platform.

MDA goals are generally portability, interoperability and reusability through architectural separation of concerns. It aims, also, to support heterogeneity of modelling languages, while providing standard representations and APIs for model repositories and other tools. Otherwise, it allows definition of machine readable application and data models which allow long-term flexibility of : (i) new implementation infrastructure that can be integrated or targeted by existing designs; (ii) integration since not only the implementation but the design exists at time of integration, production of data integration bridges and the connection to new integration infrastructures could be automated; (iii) maintenance that illustrates the availability of the design in a machine-readable form gives developers direct access to the specification of the system, making maintenance much simpler and (iv) testing and simulation since the developed models can be used to generate code, they can equally be validated against requirements, tested against various

infrastructures and can be used to directly simulate the behaviour of the system being designed.

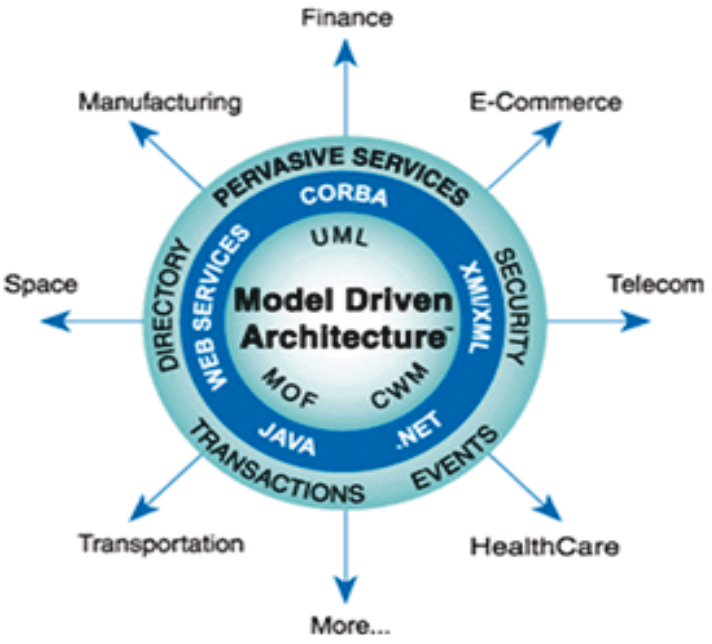

FIGURE 3.1: Model-Driven Architecture.

OMG has adopted a number of technologies, which together enable the model-driven approach. These include OMG standards such as UML, MOF, XML Meta-data Interchange (XMI) and Common Warehouse Metamodel (CWM). These standards define the core infrastructure of the MDA, and have greatly contributed to the current state-of-the art of systems modeling [80]. They have been designed to provide a general framework for defining modeling languages and corresponding UML-based graphical notations, and the facility to build editors and model repositories that have standard formats and interfaces for exchanging and interacting with models. They enable mappings from a PIM to a PSM (once a specific platform has been identified) and between a PSM and code. All OMG specifications are available on the web at [81].

3.4.3.1 Unified Modelling Language

The Unified Modelling Language (UML) is a standard modeling language for visualizing, specifying, and documenting software systems. Models used with MDA can be expressed using the UML language. UML 2 will integrate a set of concepts

for completely specifying the behaviour of objects and the UML action semantics. More information about UML specification is available at [82].

3.4.3.2 Meta-Object Facility

The Meta-Object Facility (MOF) is an OMG standard for model-driven and can be viewed as a standard to write meta-models. It provides a model repository that can be used to specify and manipulate models. Thanks to the MOF, models could be exported from one application, imported into another, transported across a network, stored in a repository and then retrieved, rendered into different formats [83]. All types of models are involved : from the structured models, those defined in UML, behavioural ones to data models can benefit from the MOF. Even the languages using non-UML can, also, participate as long as they are MOF-based. MOF is designed as a four-layered architecture. It provides a meta-meta model at the top layer, called the M3 layer. This M3-model is the language used by MOF to build meta-models, called M2-models. The most prominent example of a layer 2 MOF model is the UML metamodel, the model that describes the UML itself. These M2-models describe elements of the M1-layer, and thus M1-models.

3.4.3.3 XML Meta-data Interchange

XML Meta-data Interchange (XMI) is a model driven XML Integration framework for defining, interchanging, manipulating and integrating XML data and objects [84]. It aims, mainly, to share models using XML as it is an interchange format. In addition, XMI represents object using XML format such as XML elements and attributes. It includes standard mechanisms to link objects within the same file or across files once objects are interconnected. Like any XML file, XMI document is validate through XML Schemas. Moreover, XMI provides rules by which a schema can be generated for any valid XMI-transmissible MOF-based metamodel. It provides a mapping from MOF to XML. Basically, XMI-based standards are in use for integrating tools, repositories, applications and data warehouses.

3.4.3.4 Profiles

Profiles present a UML extension mechanism. A profile applies to a language specification, specifying a new modeling language by adding new kinds of language elements or restricting the language [85]. Consequently, the new language may be used to build a model, or by applying the new or restricted language elements to specific elements of an existing model. Any number of new profiles can be applied to an existing model, extending or restricting elements of that model. The modeler can later remove the application of a profile to a model. The result is that model as it was before application of that profile. Any model that uses a UML profile is a UML model. A model that uses a profile can be interchanged with a UML tool that does not support that profile. It will be considered by that tool as a model in UML, without the extensions of that profile.

3.4.4 Meta-modelling and Meta-Object Facility

It is necessary to understand, firstly, the concept of meta-modelling before describing the structure of the MOF architecture. If a model represents an abstraction of phenomena in the real world, the meta-model is in turn the abstraction of this model highlighting characteristics of this model to its meta-model. This one is an abstract language for some kind of meta-data. Usually, meta-models are used as a schema for semantic data that needs to be exchanged or stored; as a language that supports a particular method or process and a language to express additional semantics of existing information. A modelling language can also be defined by a meta-model that is expressed in another language called meta-modelling language.

MOF defines an abstract language and a framework for specifying, constructing, and managing technology neutral meta-models [83]. Moreover, it defines a framework for implementing repositories that hold meta-data (e.g., models) described by the meta-models. This framework uses standard technology mappings to transform MOF meta-models into meta-data APIs. This gives consistent and interoperable meta-data repository APIs for different vendor product and different implementation technologies [83]. Additionally, the MOF framework supports any kind of meta-data and allows new kinds to be added as required.

MOF has a four layered meta-data architecture inspired by the classical modelling architecture. The key feature of both the classical and MOF meta-data architecture is the meta-meta-modelling layer that ties together the meta-models and models. Basically, the classical meta-modelling architecture is based on the following four layers :

- **The information layer :** is comprised of the data to be described;
- **The model layer :** is comprised of the meta-data that describes data in the information layer. Metadata is informally aggregated as models;
- **The metamodel layer :** is comprised of the descriptions that define the structure and semantics of meta-data. A metamodel is an "abstract language" for describing different kinds of data; that is, a language without a concrete syntax or notation.
- **The meta-meta-model layer :** is comprised of the description of the structure and semantics of meta-meta-data. In other words, it is the "abstract language" for defining different kinds of metadata.

The figure 3.2 illustrates the classical four layer meta-modelling architecture. For example, the UML infrastructure is defined as a four-layer meta-model architecture. Meta-metamodeling level defines a language for specifying metamodels. Meta-model level defines the UML meta-model. Model level consists of UML models specified by the meta-model layer, and information level consists of object configurations specified by the models at model level.

The MOF meta-data architecture, illustrated by the example in Figure **??**, is based on the traditional or classical four layer meta-data architecture described above. This example shows a typical instantiation of the MOF meta-data architecture with metamodels for representing UML diagrams and OMG IDL [83]. Figure 3.3 illustrates the meta-modelling hierarchy underpinning the MDA. The MOF model is the meta-model for various modelling. It is MOF's built-in meta-meta-model. It can be seen as the "abstract language" for defining MOF meta-models. This is similar to the way that the UML meta-model is an abstract language for defining UML models.

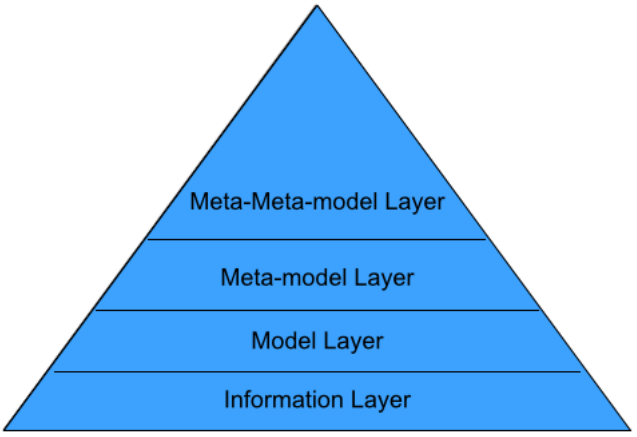

FIGURE 3.2: Four Layer Meta-modelling Architecture.

While the MOF and UML are designed for two different kinds of modelling that are, meta-data versus object modelling, the MOF model and the core of the UML meta-model are closely aligned in their modelling concepts. The alignment of the two models is close enough to allow UML notation to be used to express MOF-based meta-models. UML and IDL and meta-models at the M2 layer in turn are meta-model for models defined using the UML and IDL language at the M1 layer. They define the modelling concepts that can be used in the definition of models. The bottom level (M0) of the meta-modelling hierarchy represents instances of models, e.g. real-world entities conceptualized in terms of ontology.

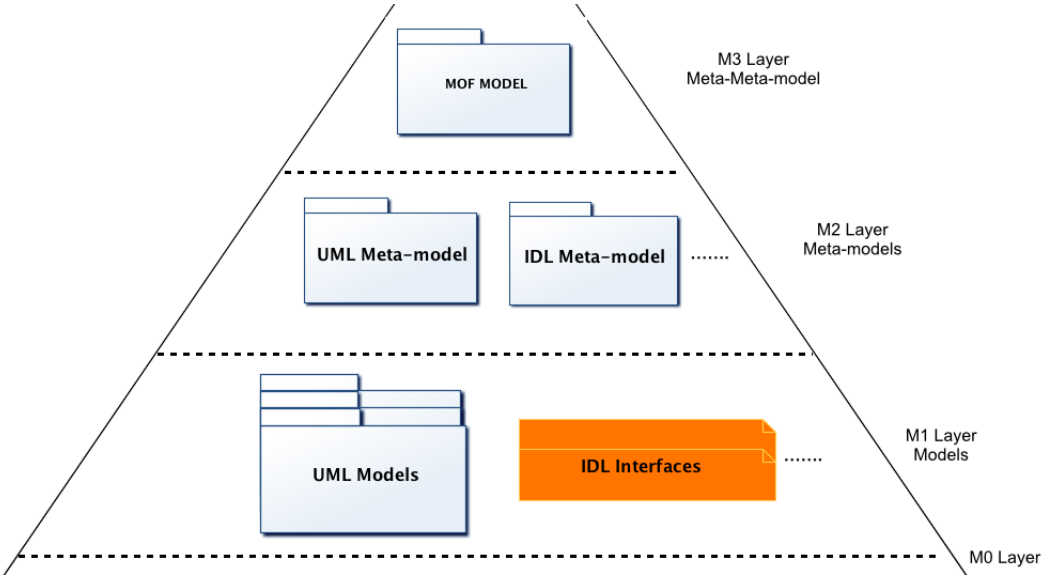

FIGURE 3.3: MOF Meta-data Architecture.

Meta-level	MOF terms	Examples
M3	meta-metamodel	MOF model
M2	metamodel, meta-metadata	UML metamodel
M1	model, metadata	UML models
M0	object, data	Modeled systems

TABLE 3.1: OMG Metadata Architecture

The table 3.1 depicts the four layer architecture of MOF framework in terms of concepts in each level and represents an example of each MOF term. The MOF supports any kind of meta-data that can be described using object modeling techniques. This meta-data may describe any aspect of a system and the information it contains, and may describe it to any level of detail and rigor depending on the metadata requirements. The term model is generally used to denote a description of something from the real world. The concept of a model is highly fluid, and depends on one's point of view. To someone who is concerned with building or understanding an entire system, a model would include all of the metadata for the system. On the other hand, most people are only concerned with certain components (for example, programs A and B) or certain kinds of detail (for example, record definitions) of the system. In the MOF context, the term model has a broader meaning [83].

What makes MOF meta-data architecture different from the classical one are some important features [83]. The MOF model is object-oriented. Meta-modelling constructs are aligned with UML's object modelling constructs. The meta- levels in the MOF meta-data architecture are not fixed. While there are typically four meta-levels, there could be more or less than this, depending on how MOF is deployed. A model is not necessarily limited to one meta-level. The MOF Model is self-describing. In other words, the MOF Model is formally defined using its own meta-modelling constructs. The self-describing nature of the MOF Model has some important consequences : (i) It shows that the MOF model is sufficiently expressive for practical meta-modelling; (ii) It allows the MOF's interfaces and behaviour to be defined by applying the MOF IDL mapping to the MOF model. This provides uniformity of semantics between computational objects that represent models and meta-models. It also means that when a new technology mapping is defined, the APIs for managing meta-models in that context are implicitly defined as well. It provides an architectural basis for extensions and modifications to the MOF model. Successive MOF RTFs have thus been able to make incremental

changes in the MOF Model to address problems that become apparent. In the future, new meta-meta-models may be added to support tasks like specification of modelling notations and model-to-model transformations. Given an appropriate set of implementation generators, it allows new MOF meta-model repository implementations and associated tools to be created by bootstrapping.

3.4.5 Ontology versus Model-Driven Engineering

We have to understand the key features of UML and Web Ontology Language (OWL) to make the best choice for privacy modelling. UML provides useful abstraction layers and different viewpoints to analyse systems. UML has a grammar for using concepts and axioms to express something meaningful within a specified domain of interest (high level or domain-specific) [86]. The grammar contains formal constraints like specifying what it means to be a well-formed statement, assertion, query, etc. on how concepts of the ontology can be used together. The translation of package to ontology is straightforward. Both UML and OWL support a fixed defined extent for a class (OWL oneOf, UML enumeration). In addition, UML has the option for binary associations to have distinguished ends which can be navigable or non-navigable. A navigable property is one which is owned by a class, while a non-navigable is not (an integer, say) [86]. OWL properties always are binary and have distinguished ends called domain and range. A UML binary association with one navigable end and one non-navigable end will be translated into a property whose domain is the navigable end. A UML binary association with two navigable ends will be translated into a pair of OWL properties, where one is inverse Of the other.

A key difference is that, in OWL, a property is defined by default as having both range and domain. A given property, therefore, can be applied to any class. So, a property name has global scope and is the same property wherever it appears. In UML, the scope of a property is limited to the subclasses of the class on which it is defined. A UML association name can be duplicated in a given diagram, with each occurrence having a different semantics. An OWL individual can therefore be outside the system in a UML model. UML has the facility of a dynamic classification which allows an instance of one class to be changed into an instance of another, which captures some of the features of Individual. However, an object

must, always, be an instance of some (non-universal) class. Both languages allow a class to be a subclass of more than one class (multiple inheritances). Both allow subclasses of a class to be declared disjoint. UML allows a collection of subclasses to be declared to cover a superclass, that is to say every instance of the superclass is an instance of at least one of the subclasses. The corresponding OWL construct brings about the superclass that instantiates the union of the subclasses, using the construct unionOf.

OWL permits a subclass to be declared using subclassOf or to be inferred from the definition of a class in terms of other classes. It, also, permits a class to be defined as the set of individuals that satisfy a restriction expression. These expressions can be a boolean combination of other classes (such as intersectionOf, unionOf, complementOf), or property value restriction on properties (requirement that a given property have a certain value – hasValue). Otherwise, UML allows the specification of behavioural features, which are essentially programs. One use of behavioural features is to calculate property values. This use has already been considered in the properties section above (derived properties). Other programs would presumably have side effects. Facilities of UML supporting programs include operations. These latters are : (1) method names; (2) responsibilities that specify which class is responsible for what action; (3) static operations that are operations attached to a class like static attributes; (4) interface classes that specify interfaces to operations; (5) abstract classes whose operations are specified in subclasses; (6) qualified associations that are programming language data structures and (7) active classes that are classes each instance of which controls its own thread of execution control. It is proposed that the Ontology Definition Meta-model (ODM) [87] omit behavioural features of UML. Moreover, UML supports various kinds of the part-of relationship between classes. In general, a class (of parts) can have a part-of relationship with more than one class (of wholes). One composition specifies that every instance of a given class (of parts) can be a part of at most one whole. Another aggregation specifies that instances of parts can be shared among instances of wholes.

Composite structures are runtime instances of classes collaborating via connections. They are used to hierarchically decompose a class into its internal structure which allows a complex objects to be broken down into parts. These diagrams extend the capabilities of class diagrams, which do not specify how internal parts are organized

within a containing class and have no direct means of specifying how interfaces of internal parts interact with its environment. Ports and Connectors model how internal instances are to be organized. Ports define an interaction point between a class and its environment or a class and its contents. They allow you to group the required and provided interfaces into logical interactions that a component has with the outside world. Collaboration provides constructs for modelling roles played by connectors. UML permits a property to be designated read-only. It, also, allows classes to have public and private elements.

MDE is an interesting approach and methodology for handling/managing privacy issues from the design stage of applications that involve social interactions not only through the web but also inside the AmI environment. The MDE methodology and its tools focus more on the principles of creating and exploiting domain models rather than providing design paradigms to code a given software. It has several benefits that allow making the design of privacy-aware systems abstract and more close to requirements that can expressed by users in high level languages according to multiple usage perspectives [88], [89]. It provides modeling tools that simplify the design of any software functionality and decrease its complexity. In addition, it helps in the easy identification of relevant elements of the system architecture and avoids the unnecessary technical details regarding implementation issues. Moreover, it facilitates the traceability of operations and guarantees the consistency and coherence of the system during the full life cycle from the design to the usage stages. The consistency and coherence control are ensured by using a common model which can be checked by formal verification tools.

In general, MDE requires semantic modelling of models. Models and their associated meta-models constitute the backbone of privacy management approach in MDE. This latter supplies several levels of abstraction. At the bottom, the physical system (referred as M0 level) corresponds to the expected application and services. The abstraction is done by the model (usually called M1 level) that corresponds to a domain modelling language. This latter is represented by a (meta) model (referred as M2 level) that is expressed in another language or a meta-modelling language which conforms to a meta-meta-model (M3 level). The appropriate aspects of modelling languages must be formalized. For instance, the UML provides a good support for modelling systems from a fixed set of viewpoints. It facilitates communication across multiple application domains. It makes it possible to train

applications designers that can work in multiple domains. However, the complexity of UML 2.0 is reflected in their meta-models. These latter constitute a problem to be understood by developers who need to understand and use them. Indeed, it is extremely not easy for tool developers to fully identify the dependencies among concepts, and to determine whether the meta-model captures all required dependencies.

Defining a meta-model for privacy that provides the relevant and necessary common concepts and vocabulary in the UML notation for managing privacy operations and policies is insufficient. The main drawbacks of using only UML concern the lack of interoperability in the tool chain of the design, traceability, knowledge extraction and verification of the models, lack of interoperability with other domains. As mentioned in the previous item, the meta-model is specific to privacy. Unfortunately, it is not possible to merge other domains in this meta-model, no verification of semantics : when we have defined a privacy rule with this meta-model, it is not possible to verify the coherence of the model. Hence, it is possible to model contradictory rules.

Ontological based meta-modelling is a binding between two theories (ontology and meta-model) that result is a vision in which meta-modelling is a process fundamentally based on ontology to describe how to build models. UML and OWL ontology languages have their corresponding meta-models. The latter have several similarities and some differences that make them mutually compliant and easy to combine. UML and OWL meta-models contain both of them at the meta level constructs to represent classes and association, and (meta) associations that show how instances or individuals of UML or OWL classes can be related to each other. OWL does not follow the clear conceptual separation between terminology (T-Box) and knowledge base (A-box) that are present in most description logics and in MOF, which distinguishes between model and information. Using OWL and UML meta-models to express the logical syntactical structures of privacy models is not sufficient and additional constructs should added to model the semantics of the models.

In order to improve the usage of the meta-model, we propose a UML profile. This UML profile allows us to add some stereotypes and constraints on UML elements. In other words, we give a UML representation to the meta-model. With this extension, we can highly improve the interoperability of the design chain by

ensuring its compliance with UML and related software generation and validation tools, allowing the easy retrieval of privacy requirements and their corresponding management techniques directly from the UML architecture. Specific tools verify the coherence of the privacy model regarding its meta-model to guarantee the compliance to requirements and regulations. In addition, constraints are added to the UML profile to be verified during the model development. We use ontology to provide a vocabulary for describing information flows and privacy management requirements.

3.5 Validation of Privacy Control Systems

Privacy validation aims at verifying privacy policies according to the application of measures. These measures are performed on a conceptual level which does not reflect technical interdependencies. Thus, measures do not provide a reliable guarantee regarding the privacy conforming behaviour of systems. The technical verification of systems can be performed by formal conformance verification which provides an automated privacy policy verification. The whole systems or parts of a system as its information flow and probabilistic system properties could be verified. Mechanisms as model checking provide the formal conformance verification of systems. Model checking mechanisms process a model of a system and test automatically whether this model meets a given specification.

In general, formal verification mechanisms are used to specify the verified constraints. Such constraints are mostly not based on standardized concepts. Also, the process to adequately create such constraints from high-level privacy policies is poorly known. Further, privacy protection mechanisms works best if they are configured/adapted to the application domain. Therefore, the optimal privacy protection criteria have to be specified according to a specific domain. Also, other solutions with a better balance between quality of service (e.g. may be influenced by preciseness of information) and degree of privacy (e.g. generalization of values) should be provided. Evaluation whether the requested/required privacy protection matches with the provided privacy protection by (1) providing the basis for the specification and verification regarding the compliance to guidelines and principles of components, the composition of components (system privacy), systems and applications; (2)

describing possibilities to derive (personal) information and defining rules for preventing those inferences.

3.6 User Privacy Control Approaches : State of the art

In this section, we aim at studying the existing works that handle with privacy management in ambient intelligence and pervasive computing environments. We consider the privacy challenges that we have mentioned in the previous chapter and they are : semantic policies, multi-domain interoperability, fine grained privacy control, management of obligations, context awareness, adaptability and conflicts management. We define, moreover, a set of evaluation criteria to analyse privacy control approaches, privacy-aware systems and tools. Such criteria are defined at the crossroad of the following properties :

- **High-level expressiveness.** If the privacy policy model provides a high level of expressiveness, the policy representation language expresses a wide range of combinations of a large set of operators in a uniform language. These combinations are expressed without changing the input specifications and without extensions to policy languages. For instance, let us consider a data storage policy of database management service hosted in domain A and a second policy that grants access to a collaborative repository service in a domain B that might use this database service among others. We consider that the access to the repository is granted only if it is authorized by both domain A and domain B policies. In policy merging approaches, the creation of the multi-domain policy that integrates domain A and domain B policies can be done only by explicitly extending all the rules of the domain B and by including rules of the domain A. Despite that, merging the policies is not allowed all the time if the cloud provider applies a strict strategy of policy isolation. The main drawback here concerns the explosion and complexity of rules, the policies interferences as well as the loss of control and autonomy over the two domain policies. Thus, the formulation of privacy restrictions and constraints are well defined through a semantic language. Consequently,

control decision on privacy policies should take into account priority levels, overriding, refinement of privileges, etc.

- **Support of different abstraction levels.** The privacy policy framework enables the creation of multi-domain privacy policy tools that offer different levels of abstraction of the access control rules of each domain. Sharing several levels of abstraction through an incremental approach should facilitate a collaborative analysis, specification, administration and an agreement on the final multi-domain privacy policy.

- **Privacy modelling.** There are different ways to design privacy models for AmI applications. Some works have used the access control techniques to build privacy policies. Other ones have used or extended existing privacy languages. We cite, also, some works based on semantic web mechanisms. So, this criterion specifies the technique used to design a privacy model.

- **Dynamic vs static privacy management.** In spite of the use of privacy model criterion, it is important to highlight the privacy management technique. Many works have used a static privacy management technique. Generally, they pointed out a static privacy policy. This one could not be modified as it is already defined in the design phase of the AmI application. Any change that will occur to privacy model involves its redefinition from the scratch. Otherwise, other works handled privacy management in a dynamic way. To fulfil their goal, many researchers opted for the semantic web and/or artificial intelligence techniques.

- **Centralized vs decentralized architecture.** We specify the kind of the privacy application architecture to determine the discipline used for privacy management. It can be for example, a middleware, a Service Oriented Architecture (SOA) or simply a peer-to-peer architecture. Normally, AmI applications are decentralized as users could access and use services any time and everywhere. However, we have found that some of privacy applications are centralized, especially, regarding privacy services management.

- **Extensibility.** It is related to the easiness and readiness of a system for eventual extensions in the future. In other words, it describes how the privacy policy model can dynamically support its evolution and the user control mode over time. For instance, adding new constructs to the policy language, adding

combination operator, import and reuse existing policies when integrating new domains with their corresponding control systems.

- **User vs system data control.** There are many challenges of assigning ownership of context information and enabling users to express privacy preferences for their own information. This approach offers a direct link between an information source and the entities that should be entitled to control the corresponding context information for privacy purposes. This ownership relationship describes the connection between an entity and the context attributes in which they have an interest in terms of privacy. With context models, ownership can be assigned to facts or objects. Ownership can also be assigned to one user or a group of users. Finally, ownership can be based on situation by applying rules associated with the fact or object types referenced by the situation.

- **Proof of concept.** A proof of concept (abbreviated POC) aims to show that privacy model is feasible. Sometimes, making a POC requires making a minimalist prototype to show that the idea could be made to work. In other words, a POC refers to a demonstration that shows how a system may be protected or compromised, without the necessity of building a complete application for that purpose. A POC can refer to a partial solution that involves a relatively tiny number of users acting in business roles to establish whether the system satisfies some aspect of the requirements. Generally, it is, often, used to describe several distinct processes with different objectives and participant roles. Indeed, the objective of a proof of technology is to determine the solution to some technical problem, such as how two systems might be integrated or that a certain throughput can be achieved with a given configuration. Users are not needed to be involved in a proof of technology. A pilot project refers to an initial roll out of a system into production, targeting a limited scope of the intended final solution. The scope may be limited by the number of users who can access the system, the business processes affected, the business partners involved, or other restrictions as appropriate to the domain. The purpose of a pilot project is to test, often in a production environment, whether the system is working as it was designed while limiting business exposure.

- **Application field.** AmI applications cover a wide range of application fields such as healthcare, social networking, academic domains, transport, business, etc. So, this criterion highlights the application domain for AmI application.

Besides, we note that few studies have supported the classification of existing privacy works in the literature. We have been referred, especially, to the work of Kurkovsky et al. [90] that has classified the privacy management techniques in pervasive computing environments. Hence, we have proposed in the following section an analysis framework for privacy-aware systems in AmI environments.

3.6.1 Privacy Policy Languages

Privacy policy languages aim at expressing the privacy controls that users want to express. Most of the privacy policy languages were designed for specific purposes with specific features and characteristics. There are, mainly, the following privacy languages : the Platform for Privacy Preferences language (P3P), Enterprise Privacy Authorization Language (EPAL) and PRIME policy language.

3.6.1.1 Platform for Privacy Preferences Project

The World Wide Web Consortium (W3C) has proposed the Platform for Privacy Preferences (P3P) language to enable service providers to post machine-readable privacy policies [91],[92]. It enables, especially, web sites to express their privacy practices in a standard format that can be retrieved automatically and can be interpreted easily by user agents. P3P is the most widely deployed privacy language, but is, also, the least expressive. To make use of P3P policies, users express their privacy preferences in a language such as "A P3P Preference Exchange Language" (APPEL) [93] or XPref [94]. The consumer's user agent, then, compares the consumer's privacy preferences with service provider's privacy promises. Service providers who post P3P policies promise specific data practices. However, P3P itself is not designed to aid policy enforcement.

P3P uses XML policy files to describe a web site's privacy practices. The XML file contains the information related to the site (name and contact), conflict management

mechanisms and the techniques of collected data In addition, there are two ways to explore the P3P language in a web site. The first one concerns the use of a P3P policy for the whole web site. The second one specifies different privacy policies for the same web site or in other words for different parts of the web site. In this case, a policy reference file is needed for specifying different policies. The example of P3P policy in figure 3.4, inspired from [91], shows some basic features of P3P.

```
1  <POLICIES>
2  <POLICY discuri="http://p3pbook.com/privacy.html" name="policy">
3  <ENTITY>
4  <DATA-GROUP>
5   <DATA ref="#business.contact-info.online.email"> privacy@p3pbook@com </DATA>
6   <DATA ref="#business.contact-info.online.uri"> http://p3pbook.com </DATA>
7   <DATA ref="#business.name"> Web Privacy with P3P </DATA>
8   </DATA-GROUP>
9   </ENTITY>
10 <ACCESS>
11  <nonident/>
12  </ACCESS>
13 <STATEMENT>
14 <CONSEQUENCE>
15    Our Web server collects access logs containing this information.
16  </CONSEQUENCE>
17 <PURPOSE>
18  <admin/>
19  <current/>
20  <develop/>
21  </PURPOSE>
22 <RECIPIENT>
23  <ours/>
24  </RECIPIENT>
25 <RETENTION>
26  <indefinitely/>
27  </RETENTION>
28 <DATA-GROUP>
29  <DATA ref="#dynamic.clickstream"> Web Privacy with P3P </DATA>
30  <DATA ref="#dynamic.http"> Web Privacy with P3P </DATA>
31  </DATA-GROUP>
32  </STATEMENT>
33  </POLICY>
34  </POLICIES>
```

FIGURE 3.4: Example of P3P Policy.

On line 2, a POLICY element is declared. A mandatory attribute of the policy element is "discuri", which must provide a link to a natural language privacy policy. This natural language policy is an important aspect of P3P. While a P3P XML policy allows for automatic negotiation, a human-readable policy should always be available to present to the user in case of doubt. This natural language policy should of course describe the same privacy practices as its XML equivalent. The ENTITY element on line 3 describes the legal entity responsible for this policy. All the DATA elements reference a data structure defined by P3P. P3P has many predefined data structures to provide not only syntactic but also semantic information about the data used in privacy policies. The ACCESS element on

line 12 indicates what kind of access the site provides to the subject of identified information. For example, a web site could offer users the possibility to change their addresses. The P3P web site [91] gives more details and information about P3P policies.

The scope of P3P is limited to the concepts of notice and choices consent that we have described in the chapter1. Users are limited to the defined web site's privacy policy. Based on this policy, users can choose to use the service or decline it. P3P does not intend to enforce privacy by technical means. In other words, user has no mechanism for policy enforcement after initial disclosure of personal information. It is up to the user to make a decision whether or not to trust the service provider. Moreover, P3P does not take into account contextual information but it is suitable to be extended and to be context-aware privacy language.

Many works have implemented several softwares based on the P3P standard. The JRC Policy Workbench [95] is an API for building policy editing and testing environments. It includes an implementation of an editor for P3P 1.1 and P3P 1.0 policies. Netscape 7.0 has introduced two new privacy-related features based on the P3P standard [96]. Users know the privacy practices of web sites with the P3P Privacy Policy Viewer. They are also informed about cookies with P3P Cookie Management. In addition, the "Privacy Bird" tool will help Internet users stay informed about what information they provide to web sites could be used [97].

Automatically, it looks for privacy policies at every visited web site. A user could specify its privacy concerns. Then, "Privacy Bird" will tell him whether each site's policies match with his personal privacy preferences by using bird icons. There are, in addition, many P3P based policy generators. For example, P3PBuilder is a P3P policy generator that creates privacy policies to the W3C specification [98]. It is an easy to use web-based generator provided with technical support to assist with installation. Another P3P based tool is P3P Display [99]. It is a free web-based utility that reads in a P3P policy file from a provided web site. It parses the contents and displays a human readable form based on the attributes in the policy and text definitions from the P3P specification. It also displays an equivalent compact policy that reflects the site policies.

In another work [49], Hong and al. proposed an extension of P3P. They defined a specification for representing user privacy preferences for context-aware applications.

Indeed, they've introduced a set of context-aware parameters to the P3P policy file like what, when, who, when and where. They described a privacy management infrastructure which could easily be plugged into the middleware. In [100], Zuiderweg studied privacy control in one particular context-aware environment: the Web Architectures for Services Platforms (WASP). This one uses Web services technology for providing context-aware services. In his thesis, Zuiderweg designed privacy architecture for the WASP platform. This one has the advantage to allow to its users the control of their privacy while being unobtrusive. It also enables expressing the types of contextual data which are relevant to WASP.

3.6.1.2 Enterprise Privacy Authorization Language (EPAL)

EPAL [101], [102], [103], [104] is a privacy language designed by IBM to enforce privacy policies within the enterprise. It aims at formalizing enterprise-internal privacy policies. Based on XACML, it defines the semantics of its policies in terms of an algorithm for evaluating policies. The authorization service interprets the EPAL policy description. Then, it responds with the set of restrictions that the policy applies on the desired action. An EPAL policy defines lists of hierarchies of data-categories, user-categories, purposes, sets of (privacy) actions, obligations, and conditions [105]. "User-categories" are the entities (users/groups) that use collected data (e.g., travel expenses department or tax auditor). "Data-categories" define different categories of collected data that are handled differently from a privacy perspective (e.g., medical-record vs. contact-data). Purposes model is the intended service for which data is used (e.g., processing a travel expenses refund or auditing purposes).

Actions model indicates how the data is used (e.g., disclose vs. read). Obligations define actions that must be taken by the environment of EPAL (e.g., delete after 30 days or get consent). Conditions are Boolean expressions that evaluate the context (e.g., "the user-category must be an adult" or "the user-category must be the primary care physician of the data-subject"). Then, these elements are used to formulate privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations. In order to allow general rules and exceptions, EPAL rules are sorted by descending precedence. For example, a rule about a

particular employee can be inserted before the rule about the department in order to implement an exception. The picture 3.5 shows a non-normative UML overview over the elements of an EPAL policy. Sub-elements are depicted as aggregation by value while references using id/refid pairs are depicted as ordinary relations. The element $< epal-vocabulary >$ and $< epal-policy >$ are the two top-level elements of EPAL. However, EPAL is more fine-grained than P3P language. It is not a context-aware privacy language but it is an interoperable language for exchanging privacy policy in a structured format between applications. EPAL represents, also, technical difficulty in representing privacy policies as a machine-readable code. A very large number of EPAL rules could not be adapted to a specific application domain which makes it difficult to implement as well as maintain.

Barth et al, in [106], have developed the Declarative Privacy Authorization Language (DPAL). It is a formal policy language that does not finish the evaluation of the policy. When interpreting a DPAL policy, the authorization service collects requirements from all applicable statements, unlike in EPAL. DPAL policies, therefore, enforce each of their statements, enabling both local reasoning and combination. Given a statement from a DPAL policy, an auditor knows the policy enforces the statement without examining the entire policy. Concatenating two policies produces a policy that enforces each statement from each policy. Therefore, in DPAL, concatenation achieves policy combination. Every EPAL policy can be translated into a DPAL policy, using conditions extended with logical operators. Translating EPAL policies into DPAL enables local reasoning and policy combination, although, combined policies might not be expressible in EPAL. Using these extended conditions; DPAL can express the same unsafe policies expressible in EPAL. However, DPAL can be restricted to express only safe policies by restricting conditions to be closed upwards. DPAL allows inconsistent policies to be expressed. However, inconsistencies can be detected algorithmically prior to deployment.

In the proposition of Wishart et al [107], contextual information owners are able to express their privacy requirements using context-dependent privacy preferences. These preferences can be defined for certain contexts. Then, they are combined to form a comprehensive privacy policy. The privacy preference language supports two preference types. The first one is the "binary privacy preferences". It allows disclosure to be either granted or denied. The second one is "granularity preferences" that enables context owners to specify detail level restrictions on the

disclosure of their context facts. Context Management System incorporates contextual information privacy mechanism's functionality. This Context Management System is able to dynamically discover new sources of contextual information, and new executable programs (which are accompanied by SensorML Process Chain descriptions) to process the information from context sources. This enables it to obfuscate contextual information without the need of detailed taxonomies.

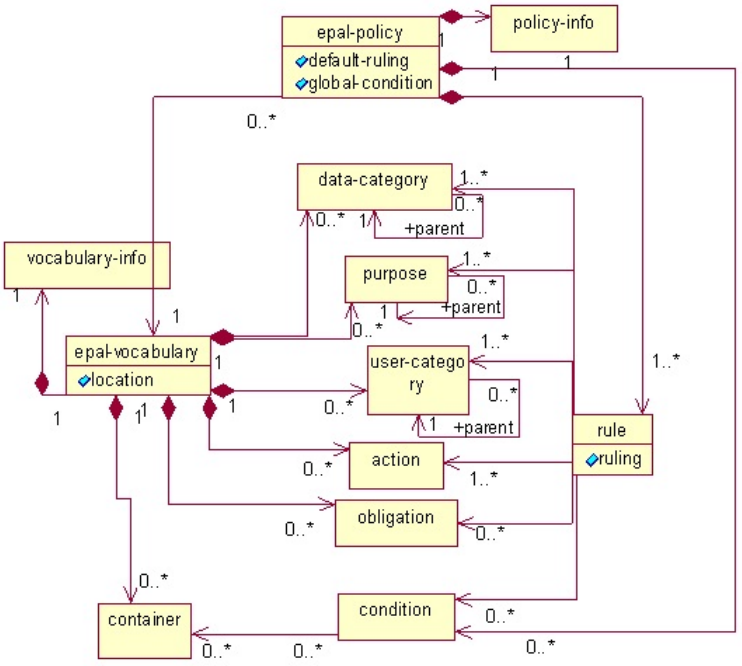

FIGURE 3.5: Non-normative High-level UML Overview of an EPAL Policy.

3.6.1.3 PRIME Policy Language

The PRIME project [108] is a large-scale research effort that aims at developing an identity management system. This one is able to protect user personal information and to provide a framework that can be smoothly integrated with current architectures and online services. In this context, an important service for helping users to keep control over their personal information is represented by access control solutions enriched with the ability to support privacy requirements. To fully address the requirements posed by a privacy-aware access control system, different types of privacy policies have been defined in the context of PRIME. Access policies define authorization rules concerning access to data or services. Release policies govern release of properties, credentials, or personally identifiable information

(PII) of the party and specify under which conditions they can be released. Data handling policies (DHP) regulate how personally identifiable information (PII) will be handled at the receiving parties. Users can define restrictions on the secondary use of their personal information which will be attached to the PII or data they protect.

A data handling policy regulates which subject can execute which actions on which resources under which conditions. Because of being tagged to a resource, the PRIME data handling policy consists of three elements: recipient, action, and restriction [109]. Recipient, the third party to which personal information can be disclosed, can be defined by an identity, a category or attributes. Action is used to denote privacy-relevant operations that recipients can require on personal data (e.g., read, disclose, modify). A privacy statement specifies restrictions that have to be satisfied before access to data is granted. Restriction can be divided into the purpose, for which the data will be used, and conditions. We distinguish between three kinds of conditions: provisions, obligations, and generic conditions. Provisions are actions that have to be performed before access can be granted. Obligations represent actions that have to be performed after access has been granted. In addition, generic conditions can be satisfied at run-time when the request is processed. Moreover, PRIME is based on several principles [108]: (i) design must start from maximum privacy; (ii) explicit privacy governs system usage; (iii) privacy rules must be enforced, not just stated; (iv) privacy enforcement must be trustworthy; (v) users need easy and intuitive abstractions of privacy; (vi) privacy needs an integrated approach and (vii) privacy must be integrated with applications. It proposes, also, an ontology for user policy which could be extensible.

3.6.2 Privacy Policies based Access Control

Access control is the traditional centre of gravity of computer security. It is where security engineering meets computer science. Its function is to control which principals (persons, processes, machines, etc) have access to which resources in the system. For example, they determine which files they can read, which programs they can execute, how they share data with other principals, and so on [110]. Access control mechanisms may express a very rich and complex security policy.

According to Sloman, policies are specified as objects which define a relationship between subjects (user, computer, process, program, etc) and targets (managed objects) [111]. An access control model designs "a class of policies with similar characteristics" which makes particular choices about what is in the protection state and how actions are treated.

Access control models are, sometimes, categorized as either discretionary or non-discretionary. The three most widely recognized models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role Based Access Control (RBAC). MAC and RBAC are both non-discretionary. These models constitute the classical ones. We also depict contextual access control models that allow taking into account dynamicity and flexibility of security privacy. An important service for helping users to keep the control over their personal information is represented by access control solutions enriched with the ability of supporting privacy requirements. In general, such access control policies specify the following elements: (1) who specifies the user identities or roles; (2) what concerns the resources or data; (3) how to specify the actions; (4) why concerns the purpose and context; (5) conditions under which allowed or denied and (6) Obligations to determine if they allowed or denied access.

3.6.2.1 Classical Access Control Models

Discretionary Access Control (DAC) restricts the access to the information based on the identity of users and/or membership in group. Access decisions are typically based on the authorizations granted to a user based on the credentials it presented at the time of authentication (user name, password, hardware/software token, etc.). In most typical DAC models, the owner of information or any resource is able to change its permissions at his discretion (thus the name). DAC has the drawback of the administrators not being able to centrally manage these permissions on files/information stored on the web server [112]. In other words, each object has an owner that determines the list of authorized subjects to access to this object. Consequently, in the discretionary mode, a subject has the control of its own objects, an owner is often the object's creator and an owner determines permissions and access rights to resources. However, there are many variants of these principles. For example, an owner can delegate to another subject access right to other objects.

Generally, DAC models are static because of access matrix could not be modified easily especially if it is big. Nevertheless, they have the advantage of having decentralized policy.

Mandatory Access Control (MAC) ensures that the enforcement of organizational security policy does not rely on voluntary Web application user compliance. MAC secures information by assigning sensitivity labels on information. It compares this to the level of sensitivity a user is operating at. In general, MAC access control mechanisms are more secure than DAC yet have trade-offs in performance and convenience to users. MAC mechanisms assign a security level to all information, assign a security clearance to each user, and ensure that all users only have access to that data for which they have a clearance. MAC is usually appropriate for extremely secure systems including multilevel secure military applications or mission critical data applications. However, MAC models are not flexible as they handle just one problem at the same time. Moreover, they don't take into account contextual parameters. So, they present static models.

With Role-Based Access Control (RBAC), access decisions are based on an individual's roles and responsibilities within the organization or user base. The process of defining roles is usually based on analysing the fundamental goals and structure of an organization. It is linked to the security policy. For instance, in a medical organization, the different users' roles may include those such as doctor, nurse, attendant, nurse, patients, etc. Obviously, these members require different levels of access in order to perform their functions. However, the types of Web transactions and their allowed context vary greatly depending on the security policy and any relevant regulations (HIPAA, Gramm-Leach-Bliley, etc.). An RBAC access control framework should provide Web application security administrators with the ability to determine who can perform what actions, when, from where, in what order, and in some cases under what relational circumstances. In [113], authors provide some great resources for RBAC implementation.

In the last years, privacy protection was the subject of several researches, particularly on privacy policies, in which different approaches for policies specification were proposed. In [114], authors addressed some of the issues related to security and privacy in pervasive computing. They explored the challenges for building security and privacy in such environments. Then, they proposed a MDW OS, GAIA, a generic infrastructure for pervasive computing environment that provides

core services to support and manage active spaces and pervasive applications within these spaces. To ensure security and privacy for pervasive application users, they proposed a dynamic security policy model. This one enables the creation of customizable programs that can be deployed to enforce and implement strong security policies that can be adapted to a changing environment. To do so, they used dynamic access policies based on RBAC model. They introduced temporal quantifiers and authorized proofs to security policies.

Otherwise, several solutions related to the access control management in Web 2.0 environments have been proposed in [113], [112], and [115]. In [116], Carminati and al. have proposed a rule-based access control model for online social networks. In their solution, social members or data owner as it's defined in their paper could define their access control rules. Digital certificates are, then, used to enforce those rules. However, the process of generating and verifying digital certificates requires a relatively high degree of sophistication from the users, which may not be appropriate in Web 2.0 settings. Compared to the work [116], a more practical but coarse-grained solution for enforcing social relationship was proposed by Mannan and van Oorschot [117]. Their idea is to leverage the existing circle of trust in Instant Messaging (IM) networks. Gates, in [118], re-examined privacy protection notion in social networks. He considered that relationship based access control is a new security paradigm addressing the requirements of the web 2.0.

3.6.2.2 Contextual Access Control Models

Since the late 1990s, with the development of Internet-based distributed systems, a new access control model - the Attribute Based Access Control (ABAC) - has become increasingly important. In ABAC, access decisions are based on attributes of the requester and resource. Users don't need to know the resource before sending a request. Current research and development efforts of ABAC, usually, focus on one kind of policy definition. However, it cannot support multiple policies. Hence, in order to establish an authorization mechanism suitable for Grid computing, further research is needed. ABAC uses attributes as building blocks in a structured language to define access control rules and to describe access requests. Attributes are sets of labels or properties which can be used to describe all the entities that must be considered for authorization purposes. Each attribute consists of a

key-value pair such as "Role=Assistant Manager". Axiomatics [115] implements ABAC using XACML and it considers these four entities: (i) the subject who is demanding access to an information asset; (ii) the action which the user wants to perform; (iii) the resource identifying the information asset or object impacted by the action and (iv) the environment identifying the context in which access is requested.

Temporal-RBAC (TRBAC) is an extension of RBAC models. It supports temporal constraints on the enabling/disabling of the roles. TRBAC supports periodic role enabling/disabling, and temporal dependencies among such actions. Such dependencies expressed by means of role triggers (active rules that are automatically executed when the specified actions occur) can also be used to constrain the set of roles that a particular user can activate at a given time instant. The firing of a trigger may cause a role to be enabled/disabled either immediately, or after an explicitly specified amount of time. Enabling/disabling actions may be given a priority that may help in solving conflicts, such as the simultaneous enabling and disabling of a role. As expected, the action with the highest priority is executed [119]. However, TRBAC does not distinguish between a role being enabled and a role being active. A role is enabled if the temporal conditions associated with it are satisfied. It is active if a user has logged in the role and only enabled roles can be activated. Because of such limitations, TRBAC cannot support some forms of constraints, such as the maximum number of activations of a role by a user in a given time interval.

Generalized-TRBAC (GTRBAC) extends TRBAC by introducing temporal conditions on user-role assignments, especially, on role-permission assignments. For example, "there are at most ten users activating the role "DayDoctor at a time". GTRBAC takes into account each of the relations that joins the entities in RBAC of it. It allows overloading these relations by temporal constraints. Thus, we can constraint the activation of a role, subject assignment to a role and the assignment of permission to a role. However, this multiplication of the constraint involves some conflicts. Besides of the activations, arbitrary of the releases associated to the constraints of interdependence between these releases can create some ambiguousness. One can, easily, create a buckle while binding two releases between them. GTRBAC has, now, the possibility to give some priorities to the rules of activation. However, every type of constraint, also, possess its own type of hierarchy. In order

to solve the incoherence, a dependence graph, is established. This one verifies the consistency thanks to the pre and post conditions of the constraints. All these mechanisms return the expression of GTRBAC complex. Thus, one notices that, although, bidder more of possibility to its user, the heap of the different types of constraints generates a complexity in the expression of the model. In the continuation of their works, Joshi and Al. year of simpler models they propose an algorithm [120] to replace the relations of user affectation to the roles by temporal roles. Thus, they come back to a model more flexible and less complex.

GeoRBAC is an RBAC model that introduces geographic location as a further constraint on role activations [121]. In particular, it represent objects and assigns spatial extents to roles. This model describes how locations on Earth are represented in GEO-RBAC. Objects are embedded in the Euclidean space E whilst a spatial reference system maps locations in E onto places on Earth. They have a geometric representation (geometry) compliant with the OGC (Open GeoSpatial Consortium) simple feature geometric model [122]. In such a model, the geometry of an object can be of point, line or polygon type, or recursively be a collection of disjoint geometries. A point describes a single location in the coordinate space. A line represents a linear interpolation of an ordered sequence of points. A polygon is defined as an ordered sequence of closed lines defining the exterior and interior boundaries of an area. An interior boundary defines a hole in the polygon. Resources to be protected consist of data about entities of the real world that may occupy a position. To be compliant with the OGC terminology, these entities are called features [122]. These latters are identified by names. Features are spatial when entities can be mapped onto locations in the given space. The location of a feature is represented through geometry. Conversely, features are non-spatial when they are not associated with any location. Feature location is formally defined as the triple of "Feature location", "Feature functions" and "Feature type ordering". Objects in GEO-RBAC can be extensionally represented by listing the features belonging to the set or by intentionally specifying a query either spatial or non-spatial over a feature type extension. The object in this case corresponds to the query result.

OrRBAC model provides rich panel of security modeling features [123]. The application concept of context is more open. OrBAC seems to be suitable to build an access management in multi domain context where we deal with concepts like domain, organization and access. But the limitation affecting this model is the

lack of semantics and security rules support. From a pragmatic point of view, the security service should consider the factor of domain, time, location and trust to enhance the security of interaction between each other. It is important to note that each RBAC extension is not a complete security solution in itself. It is highly interesting to integrate a selection of RBAC extensions, depending on the contextual parameters we want to take advantages from and build a prototype model based on RBAC extensions.

In another work [124], authors have proposed an approach built upon Or-BAC to define a contextual security policy that will be applied to the information system. This enables the definition of multiple equilibrium points between security, performance, convenience and compliance objectives. These equilibrium points are expressed as contexts or context combinations of the security policy. Indeed, the Or-BAC framework includes tools for formally verifying the security policy and for translating the formal security policy into practical configuration scripts that can be applied to policy enforcement points to change the security policy. The expression of the security policy allows the definition of simple responses to each threat, a global and efficient response in the face of multiple threats being computed during the instantiation of the security policy. The developed method has the advantage to help the administrator in updating the policy. Other contexts must be defined to specify additional security rules to be triggered when intrusions are detected. In fact, a parallel could be drawn with provisional authorizations [49]. Contexts are linked to the history of reported intrusions, and activate provisional security rules. Some of these security rules may correspond to permissions (positive authorizations) but more often they will represent prohibitions (negative authorizations). The prohibitions will be automatically deployed over the information system as a reaction to the intrusion. For instance, this may correspond to automatically insert a new deny rule in a firewall.

Context-Role based access control (CRBAC) is a new model supporting simultaneously several dimensions: time, location, trust, etc. Context means situational information. Almost, any information available at the time of an interaction can be seen as context information. It can be an identity, spatial information (e.g. location, orientation, temporal information like date and season of the year), environmental information (e.g. temperature, social situation such as behaviour, resources that are nearby, availability of resources). Along with the traditional concepts of RBAC

like users, roles, permissions, CRBAC proposes a new concept called Context-Role (CR) that represents a set of context roles. The context role is used to capture security-relevant context information about the environment for use in CRBAC policies. The context role can contain time-related context role, location-related context role, etc. The context role shares many characteristics with user roles. So, context role has role activation, role revocation, and role hierarchies.

In another work [125] [126], [127], Qun et al. introduced a family of models (P-RBAC) that extend RBAC model in order to provide full support for expressing highly complex privacy-related policies, taking into account features like purposes and obligations. In their model, referred to P-RBAC, privacy policies are expressed as permission assignments (PA). These permissions differ from permissions in classical RBAC because of the presence of additional components, representing privacy related information. They also developed conflict analysis algorithms to detect conflicts among PA. Thus, this avoids the problems that EPAL [128] rules have because of its sequential semantics [106].

Hart et al, have proposed in their work [129] a new approach for Content Based Access Control (CBAC). This approach aimed to reduce user work as user doesn't have to specify access rules per document. Tiny policies, based on CBAC, are defined. This has the advantage to facilitate the understanding of policies and their modification. The CBAC policies are also expressive. They are able to capture many levels of granularity. In [130], Bags et al. presented a privacy manager interface for non- expert users to interact with their developed User-Centric Privacy Framework (UCPF). Users are, then, able to define and administrate their privacy preferences. The UCPF is introduced as a novel mechanism to enable personal privacy for the inhabitants of the smart home. The first prototype is based on access controls techniques. In [131], Sheikh et al. have introduced the notion of Quality of Context (QoC) information for context management middleware. Five QoC indicators have been proposed and different options available for heir quantification have been discussed. Users' privacy is protected through a QoC-based privacy policy framework while privacy policies are defined with GeoPriv Common Policy format [132]. In [133], Cornwell et al, have given an overview of their work in developing some core technologies for helping end-users manage their security and privacy. They aimed to provide simple user interfaces and visualizations for

specifying and understanding policies. They have also described three applications to evaluate these technologies.

3.6.2.3 Usage Control Model (UCON)

Usage Control (UCON) [134] is a new access control model that extends and goes beyond traditional trust management, digital rights management and access control models by integrating obligations and conditions, as well as authorizations continuity and the strategies of attributes mutability in covering security and privacy. Therefore, usage control policy allows systems to enforce the security before the access request, during the use and after the services use. Usage control allows, in fact, an efficient tractability of services usage in a multi-domain open environment such as the cloud, where domains are loosely coupled and managed independently. Indeed, UCON policies are defined using eight concepts namely: subjects, subject attributes, objects, object attributes, rights, authorizations, obligations and conditions. The authorization, obligations and conditions are components of usage control decisions. The authorization consists of deciding whether to permit a particular form of service and data use. Normal authorization decision can be either permit or deny based on subject and object attributes and conditions. Conditions are system environment and context restrictions that are not explicitly related to subject or object attributes.

Obligations are the actions that should be performed by subjects or by the access control system. Unlike traditional access control models such as RBAC or MAC that are applied only on service-side, UCON model is applied on both service provider and consumer sides, in order to guarantee a persistent control, during the usage time and even after. Consumer-side control requires the existence of a trusted computing base and a reference monitor. Moreover, UCON can be applied for systems with centralized or decentralized access control such as peer-to-peer systems. Moreover, RBAC model requires that all domains users must be already known by the resource a priori (i.e., user should have accounts that are provisioned through roles), whereas the UCON model does not require from users to be known by the resource a priori. Therefore, the UCON model can cope better with highly distributed environments; this is also because of its support of attributes mutability.

3.6.2.4 Extensible Access Control Markup Language

Extensible Access Control Markup Language (XACML) [69] is the result of OASIS standardization effort. It proposes an XML-based language to express and interchange access control policies. In addition to the language, XACML defines both architecture for the evaluation of policies and a communication protocol for message exchange. In a typically scenario, a user wants to perform some action on a resource. Therefore, he/she issues a request to the device protecting the resource which is called Policy Enforcement Point (PEP). The PEP creates a request which consists of four attributes: (1) the Subjects (or users) who make the request, (2) the Resource that will be accessed, (3) the Action that has to be performed on the resource and (4) the Environment which is an additional information related to the request. The PEP sends this request to the Policy Decision Point (PDP) which processes the request by looking for some policy that applies. It sends the answer back to the PEP which permits or denies access to the user.

XACML defines four layers to access policy control as seen in the figure 3.6. The first layer is the Policy Administration Point (PAP) one. It creates security policies and stores these policies in the appropriate repository. Policy Enforcement Point (PEP) is the second layer. It performs access control by making decision requests and enforcing authorization decisions. The third layer concerns the Policy Information Point (PIP). This one serves as the source of attribute values, or the data required for policy evaluation. Policy Decision Point (PDP) is the last layer. It evaluates the applicable policy and renders an authorization decision. The different elements of the policy elements of XACML are policies or policy sets, combining algorithms, obligations, targets, rules, attributes, attribute values, functions, and effects. An XACML policy represents a single access control policy, expressed through a set of rules. It contains a single target, 0 or more rules, 0 or more obligations, and a rule combining algorithm.

Each combining algorithm represents a different way of combining multiple decisions into a single decision in case of different access control decisions through multiple policies or rules. There are seven standard algorithms, for example the deny overrides algorithm which effects that if any evaluation of a rule returns deny, then the final result is also deny. Obligations will be passed back in the response from the PDP. There are additional operations which the PEP should perform when

enforcing the authorization decision (e.g. provide notification to a customer after a wire-transfer operation has been performed by the bank). A Target is a set of conditions that must be met for a policy or rule to apply to a given request. There are conditions for a subject, a resource, and an action.

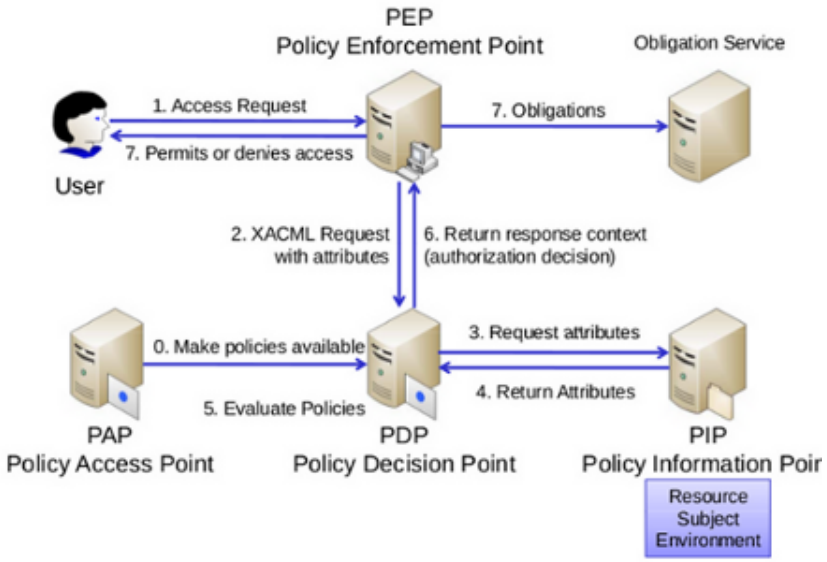

FIGURE 3.6: XACML Architecture (simplified).

Otherwise, since version 2.0 of XACML, there has been the possibility to use profiles which are kinds of guidelines for the formulation of policies and specify the use of XACML in specific scenarios. The goal is the use standard elements, attributes, and functions, and therefore to prevent errors, because all aspects of the scenario are regarded. One predefined profile is the privacy policy profile. It adds two values for the specification of a purpose for which the data was collected and for which an access to the data is requested.

Another approach pointed out by Trabelsi et al. [135] in the case of service discovery in SOA applications. It is a registry based solution in which context-aware security policies are enforced in order to ensure privacy and access control for clients and services. Users are able to specify their security preferences that will be enforced during the discovery process. Authors have extended WS-Discovery to incorporate appropriate confidentiality and privacy protections restricting the potential matching between a client lookup request and a service profile. Discovery policy specification is composed of a set of rules in XACML to control the access of sensitive resources (services profile). Regarding privacy protection, the client can protect his private information (identity, intentions, favourite services, etc).

Moreover, a user should be able to protect his personal information such as his health status or his medical history.

3.6.3 Ontology based Privacy Policy Languages

Ontologies are proposed as an alternative or add-on framework to XML-based policies to overcome their low expressiveness and lack of formal semantics regarding security and privacy management. Rei [136], KoAs [137], Protune [138], ROWLBAC [139], OWL-POLAR [140] are good examples of ontology-based access control policy languages with a higher level of abstraction, interoperability and rich semantic expressiveness for the management of access control and resources in distributed information systems [141]. These languages, usually based on semantic web and logic programming, are interesting candidates for defining multi-domain policies for collaborative cloud services. Indeed, they allow different software agents to understand policy terms and to enforce these policies as intended by their semantics. In this section, we discuss how these languages can overcome XML-based limitations and address high-level semantic interoperability requirements.

KAoS is a policy representation language based on OWL where policies are expressed as OWL ontologies by extending four types of policies namely: Positive Authorization, Negative Authorization, Positive Obligation, and Negative Obligation [137]. Policy representation in KAoS can be made at different levels of abstraction using elements such as groups, actors, action properties and conditions of applicability. Each policy is associated with an action class that represents the collection of events, with similar nature, and that is performed by a given actor (a managed element in the managed system). In addition, policies can be associated with priority and relation constructs to facilitate policy grouping, conflicts management and dynamic inference of relations among policies. KAoS is also a policy management framework that provides interesting tools such as: policies creation, edition, querying and administration. It also allows storage policies, conflicts detection and resolution. For example, KAoS Guard is a policy decision point that allows inferring if the policy enforcement and disclosure mechanisms are applicable in a given situation.

Similar to KAoS, Rei is a policy management framework intended to provide a declarative, simple and flexible policy language for privacy and security [136]. Rei

policies are described at different levels of abstraction using attributes of users, actions and other contexts instead of identity attributes [136]. Inspired from human policies, Rei policy language includes sanctions and conditional permissions that refer to the consequences when a request deviates from a policy. For instance, a conditional permission grants an entity with a permission to perform actions only if this entity gets certain additional responsibilities. Conditional permissions impose additional obligations on the entity after the authorization. Rei policies are expressed using OWL-Lite (or RDF-S), over domain specific ontologies represented using RDF, DAML+OIL and OWL. Rei policies are defined as sets of constraints over authorized and obligated actions. They are based on the concept of deontic object that is expressed using the following concepts [142]: rights, prohibitions, obligations and dispensations. The Rei policy language also includes six speech act concepts which are primarily used for defining policy management actions such as: delegate, revoke, request, cancel, command and promise. For instance, the delegate speech act is interpreted as the creation of a new permission on an object for a subject, while the revoke speech act removes from the policy the existing permission assertions and therefore, implies a prohibition decision. The transfer of permissions from one entity to another is done through the request speech act concerns. It is usually implemented as the delegation of an action execution on behalf of the requester. The command speech act causes an obligation on the recipient while the promise speech act implies an obligation on the sender [142].

Moreover, any Rei policy can also be customized through the inclusion of context types in the ontology domain. For instance, if there is a need to model the action that consists of reading a file in a given directory, the general action class can be extended with properties about the directory like the modification date, etc. Rei adopts both ontology and rule-based reasoning to express semantically-based policy enforcement rules. Ontology reasoning allows policy querying. The Policy reasoning engine is implemented in Java and uses Prolog. Rei security model is different from RBAC because it does not support the assignment of roles or permissions directly to a subject. Certainly, it does not distinguish between Role-based and Group-based policies allowing them to be described using the same set of constructs, leading to simpler policies. While KAoS is able to detect policy conflicts statically, Rei includes the Metapolicy concept to manage dynamically the conflict of policies. This is managed by setting a modality preference that can be negative/positive or vice versa. It can also be managed by stating the priority between rules within

a policy or between policies themselves. Separation of duty conflicts, using Rei conditions, is difficult to handle and the conditions are dependent on the context and the application domain. Rei is a high level security and privacy usage control language that can manage delegation and obligation. However, the enforcement of Rei policies remains unclear, since it is limited only to Prolog.

Protune is a flexible policy language used for expressing security and privacy policies, business rules and trust management policies, as a logic program that can be written in object oriented syntax [138]. The concept behind Protune is that security systems can negotiate access rights to resources, in a peer-to-peer manner, by iteratively exchanging only the minimum required information and credentials, using only two predefined predicates: credential and declaration. This approach allows preserving the privacy of each peer security system. At the end of the iterations, a peer can generate natural language explanations describing the negotiation's decision such as why the access is granted or denied.

Tim Finin et al [139] proposed the ROWLBAC ontology model with the aim to define access control policies compliant to the RBAC security model. They proposed two approaches for modeling roles: roles as classes or roles as property values. In the first approach, each RBAC role is represented by two OWL classes: $< Role >$ and $< ActiveRole >$, describing the static assignment of the role and the dynamic activation of the role, while the role hierarchy is represented using the class hierarchy. This approach directly supports the static separation of duty and the dynamic separation of duties via the OWL property construct "disjointWith". In the second approach, roles are modeled as instances of a generic class, named Role. The main limitation of this model is that it does not provide a schema for role definition, according to the open world assumption of OWL ontology. Besides, it imposes the creation of new individual classes, as many times as roles are needed. This is not practical in the case of a large organization with a large number of users, functions and access rights. Similar to other ontology languages, ROWLBAC allows the extension of the core ontology with additional domain dependent ontologies to describe roles, resources and actions for a given application domain. In ROWLBAC, it is possible to detect conflicts (segregation of duties) using OWL classes and properties.

OWL-POLAR [140] is an OWL-DL based policy language for representing semantic policies. It supports a decidable policy analysis and provides reasoning mechanisms

that allow the anticipation of possible conflicts between policies. Activation conditions of OWL-POLAR policies are defined using semantic conjunctive formulas which allow variables to be used in defining policies. However, OWL-POLAR allows only object variables to be compared, using only two OWL class properties constructs, namely owl:sameAs and owl:differentFrom whereas the OWL data-type variables can be used in defining constraints on the data-type properties. OWL-POLAR converts the reasoning on the security policy into concept subsumption tests or query answering operations. Policy modeling is based on the use of conjunctive semantic formulas which can be trivially converted and evaluated using SPARQL queries [143] and OWL-DL reasoning [144]. Policy conflicts detection relies on monotonic reasoning. This is considered as a limitation because any addition of factual knowledge to the policy will never cause this to become false while in reality conflicts detection requires non-monotonic reasoning. In order to anticipate conflicts between two policies the recovery procedure creates a canonical state of the world where these two policies are active at the same time. Thus, the standard consistency-checking operation of the Pellet OWL-DL reasoner [145] can be used to test the possibility of such a state.

Dibyajyoti et al. [146] propose a privacy-aware system that embeds semantically rich policies based on device context in the smart-phone' framework. These policies are, especially, based on Rei language. A privacy control module has been implemented to protect user privacy by performing reasoning over the context. It deals with the resource to be protected, the owner of a resource and the requester who wants to access it. Owner's profile information and the group information could be used by this module along with specified privacy policies. It enforces owner's privacy policies using static information about the owner as well as dynamic information observed and inferred from her context. It consists of (i) a set of ontologies for describing activities/context, policies and access requests, (ii) the knowledge about the owner, (ii) the privacy preferences, and (iv) a reasoning engine that accepts requests and performs the reasoning.

In [147], authors have been proposed a model that addresses the privacy concerns in a multi-user and multi-database owner environment. This model provides an assurance where by database owners are able to trust the assurances of users by making use of various audit components of the model. In addition, The model describes the key concept of segregating access to data used for processing from

access to data needed for final end use. The model uses the concepts of a mediator machine capable of reading machine -interpretable privacy policies and enforcing them through critical components like, Query manipulator, Compliance Screen and reasoning engine. The model also use the audit component consisting of a justification mechanism to check the correctness of inferences drawn by machine relating to access decisions.

In [147], authors have proposed a model that addresses the privacy concerns in a multi-user and multi-database owner environment. This model provides an assurance where by database owners are able to trust the assurances of users by making use of various audit components of the model. In addition, The model describes the key concept of segregating access to data used for processing from access to data needed for final end use. The model uses the concepts of a mediator machine capable of reading machine -interpretable privacy policies and enforcing them through critical components like, Query manipulator, Compliance Screen and reasoning engine. The model also uses the audit component consisting of a justification mechanism to check the correctness of inferences drawn by machine relating to access decisions.

In [148], Pramod et al. present a framework to provide users with appropriate levels of privacy to protect the personal information on their mobile devices. They use semantic web technologies to specify high-level, declarative policies that describe user information sharing preferences. They show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems. The privacy mechanisms constitute a baseline that can be extended and incorporated by any of the existing social networks including location based mobile social networks.

In [150], authors address the privacy-utility tradeoff by providing safe access to search logs, instead of releasing them. They propose a policy based safe interactive framework built on semantic policies and differential privacy to allow researchers access to search logs, while maintaining the privacy of the users. Semantic policies are used to infer the higher levels of information that can be mined from a dataset based on the fields accessed by a researcher. The accessed fields are then used to build research profile(s) that guide the amount of privacy to be enforced using differential privacy.

In [151], authors propose an approach for privacy management that uses semantically rich policies and reasoning mechanisms on user context. The context realized as a dynamic knowledge-base of RDF triples Is grounded in an ontology expressed in the semantic web language OWL. All policies are encoded in form of SWRL rules and use conjunctions of facts in the context knowledge-base in their conditions.

3.6.4 Analysis Framework for Privacy-aware Systems

In this section, we aim at classifying privacy-aware systems and approaches that handle privacy management in AmI. Basically, these systems are based on privacy policy languages, access control languages and ontology languages that we have presented previously. We note that few studies have supported the classification of existing works in the literature. However, we have been referred, especially, to the work of Kurkovsky et al. [90] that has classified the privacy management techniques in AmI environments. Hence, we have proposed an analysis framework for privacy-aware systems in AmI environments. This framework is illustrated in the tables 3.2, 3.3, 3.4. They contain the evaluation criteria that we have described above and they are used to analyse privacy models, systems and tools. We denote by "+ " whether the approach or language meets the criterion in question and by " - " whether it does not comply.

As we notice in the table 3.2, among the privacy policy languages, P3P and EPAL offer the more adaptability to user privacy preferences than PRIME language. All of these languages are XML based and do not correspond to our defined privacy challenges and requirements. Interoperability is not guaranteed by them. They don't take into account the contextual and situational information. They are not fine grained control. They are not expressive enough and they don't use semantic languages for privacy modeling. Among privacy policy based access control languages in the table 3.2, we underline that DAC, MAC and RBAC have, approximately, the same characteristics regarding the interoperability, context and situation awareness, semantic privacy modeling and all the privacy challenges. ABAC respect the context awareness as the policies can use any type of attributes (user attributes, resource attribute, etc.).

TABLE 3.2: Framework resuming main characteristics of Privacy Policy & Access Control Languages.

	Privacy Policy Languages			Privacy Policy based Access Control (part1)			
	P3P	EPAL	PRIME	DAC	MAC	RBAC	ABAC
Interoperability	-	-	-	-	-	-	-
Fine grained privacy control	-	-	-	-	-	-	-
Context and situation awareness	-	-	-	-	-	-	+
Semantic privacy modeeling	-	-	-	-	-	-	-
Support of unknow policies	-	-	-	-	-	-	-
Conflict management	-	-	-	-	-	-	-
Adaptability	+	+	-	-	-	-	-
History management	-	-	-	-	-	-	-
Management of obligations	-	-	-	-	-	-	-
High-level expressiveness	-	-	-	-	-	-	-
Support of different abstraction levels	-	-	-	-	-	-	-
Privacy representation technique	XML	XML	XML	XML	XML	XML	XML
Dynamic vs static privacy management	static	static	static	static	static	static	static
Extensibility	+	+	+	+	+	+	+

The table 3.3 depicts the other context-based access control languages such as TRBAC, GTRBAC, GEORBAC and ORBAC. These languages are interoperable but they are not fine grained control. They are adaptable enough to the user privacy preferences. Even, UCON and XACML present the same features. The table 3.4 shows the advantages of ontology-based policy languages and their shortcomings. These ontology-based policy languages offer semantic privacy modeling techniques that allow a minimum of interoperability. They enhance, also, the expressiveness of privacy policies. They support some levels of abstraction. Otherwise, we notice that all of the languages depicted in the three tables are extensible.

Otherwise, there were many attempts for user privacy protection in AmI and

TABLE 3.3: Framework resuming main characteristics of privacy policy based access control.

	Privacy Policy based Access Control (part2)					
	TRBAC	GTRBAC	GEORBAC	ORBAC	UCON	XACML
Interoperability	+	+	+	+	+	+
Fine grained privacy control	-	-	-	-	-	-
Context and situation awareness	+	+	+	+	+	+
Semantic privacy modeeling	-	-	-	-	-	-
Support of unknow policies	-	-	-	-	-	-
Conflict management	-	-	-	-	-	-
Adaptability	+	+	+	+	+	+
High-level expressiveness	-	-	-	-	-	-
Support of different abstraction levels	-	-	-	-	-	-
Privacy representation technique	XML	XML	XML	XML	XML	XML
Dynamic vs static privacy management	static	static	static	static	static	static
Extensibility	+	+	+	+	+	+

pervasive computing environments. Some of these works have extended existing privacy languages by adding some concepts or elements. For example, in [40], the proposed privacy model is based on the P3P language. A prototype of a part of the privacy architecture was implemented for the evaluation of context-dependent preferences. While in [152], privacy model is based on the P3P language to specify user privacy preferences. At the same time, the model is represented by ontology. We consider that the corresponding application due to the use of ontology. In other words, ontology offers an abstract level that ensures interoperability. We notice that privacy model based P3P is, almost, oriented Web application architecture. In addition, in most cases, authors proposed a static privacy management technique. Basically, the design of privacy is done just one time in the design phase of pervasive application.

However, users' needs, profiles, contexts and situation are constantly changing.

TABLE 3.4: Framework resuming main characteristics of ontology-based privacy policy languages.

	Ontology-based privacy policy languages				
	KAOS	REI	PROTUNE	ROWLBAC	OWL-POLAR
Interoperability	+	+	+	+	+
Fine grained privacy control	-	-	-	-	-
Context and situation awareness	-	-	-	-	-
Semantic privacy modeeling	+	+	+	+	+
Support of unknow policies	+	+	+	+	+
Conflict management	-	-	-	-	-
Adaptability	+	+	+	+	+
Management of obligations	-	-	-	-	-
High-level expressiveness	+	+	+	+	+
Support of different abstraction levels	+	+	+	+	+
Privacy representation technique	OWL	OWL	OWL	OWL	OWL
Extensibility	+	+	+	+	+

This rapid change makes researchers facing a great issue of privacy which is its dynamic character. Other works have focused on the access control mechanisms. For example, in [39], privacy model is based on the RBAC model and obfuscation rules. Privacy management technique is dynamic thanks to semantic management techniques or what is called obfuscation rules. An e-Wallet was developed in the context of My Campus project [42]. A case study "restaurant concierge" is implemented to illustrate the use of the e-Wallet. The academic domain is the application field of "My Campus" [42] project. It is an interoperable application thanks to the SOA architecture.

In [153], both of context and situation information are used in the pervasive application. Context is classified into physical context (obtained from various sensors, devices, actuators, and other smart objects that are distributed in the environment.) and logical context relationship that the user has with the environment and

other entities such as social relationship). Two approaches are used for assigning privacy weights to the context elements: User centric and System. Additionally, a Context-Privacy Graph (CPG) is created with the privacy setting of the system as the root and the context elements at the lowest level. It could be considered as dynamic. Dynamic rules are modeled using the knowledge of user activity and behaviour.

Moreover, vocabulary and situation grammar are used to generate user-specific dynamic rules. The methods used include systematically surveying literature and available information, designing and implementing prototypes to prove the feasibility of the proposed ideas, creating models and concepts that generalize what was learned from the prototypes, and evaluation of the proposed solutions. All interactions between users and environment are stored in the pervasive application. Context-aware rule sets are developed using the JBoss Drools rules engine [41]. Hence, the privacy set rules could not be interoperable with another privacy model.

In [10], the focus is on the location provided by sensors. It is ubiquitous sensor network architecture. However, we distinguish two kind of architecture: Web architecture and badge system architecture. There is no information about the privacy model unless 4-digit hex for the user identifier ID and the privacy preferences. The privacy control is ensured by RF beacon which is considered as a static privacy management technique. An active badge system was implemented and tested as a pervasive sensors network. There are two application fields: privacy-aware social networking and interactive media system throughout the MIT Media Laboratory. System leaves all the control to the users with their active privacy badges.

In [7], privacy preserving is not ensured by a privacy policy in this work but by a "k-means" clustering protocol. The four privacy preserving used approaches are knowledge hiding; data perturbation and obfuscation; distributed privacy reserving data mining; privacy aware Knowledge sharing. A set of consumer data are used for customer segmentation. The focus is on the privacy preservation on the use of segmentation protocol. Many computational assignments are conducted. The scale factor is linear with the normal k-means. We can consider the application interoperable as the privacy is handled with the "K-means" algorithm.

In [37], the targeted platform for the implementation of these components has been the Micro Edition of the Java 2 platform, with the Connected Device Configuration,

the Foundation Profile, and the Personal Profile. Authors have, also, evaluated privacy. Context manager stores contextual information in a data base. The implementation includes plug-ins for socket-based communication and RSAAES cryptography. Consequently, we can consider that the application could be interoperable. It is, also, extensible applications as plug-ins are used to extend the functionality of the context manager.

3.7 Conclusion and Thesis Contributions

In this chapter, we have studied the existing solutions and approaches for managing user privacy in the AmI applications. We have presented the main policy languages including those dedicated to both security and privacy. Through tables 3.2, 3.3 and 3.4, we have summarized the analysis of the different approaches according to the several criteria introduced through the challenges and requirements presented previously. Each criterion is a fundamental parameter to define privacy in AmI.

Our thesis brings an extension of the traditional system-centric approach of managing privacy in Ambient Intelligence towards a more user-centric approach that is based on policies which can be defined, customized and handled by the users themselves.

The main contribution of this thesis is the proposition of a new framework for privacy by design that allows better defining and handling privacy policies and control procedures from the design stage of AmI applications. For this purpose, we have adopted a Model-Driven Engineering (MDE) approach and proposed a meta-model for specifying and implementing privacy policies. The conceptualisation constructs of the meta-model are also based on ontology language constructs to bring more expressiveness and allow for the formal description and reasoning on privacy policies according to the closed world assumption. This combination is important, on the one hand, for enabling the full interoperability of privacy controls and policies between the different applications and, on the other hand, for offloading application designers from implementing privacy policy management operations.

The proposed meta-model also allows defining privacy policies that can be tailored to individuals' privacy needs and brings a fine-grained access control over the

private resources. The proposed meta-model simplifies the definition of privacy policies by proposing a restricted set of upper concepts, which have roots in the Ontology Definition Meta-model (ODM) and Business Process Definition Meta-Model (BPDM). Privacy policies can be defined according to specific templates that allow the controlling of the release and the handling of private ressources and the policies themselves as well as the observations of sensors in private regions. OCL constraints are defined to bring model-theoretic semantics of the classes definition in the meta model to avoid the definition of wrong privacy policies.

The second contribution of this thesis is a generic middleware for implementing the privacy management models as inference rules by using the concepts of the meta-model according to (one) the proposed policy templates. These rules can be handled by forward-chaining inference engines. A proof of concept dealing with privacy in ambient assisted living and social interactions through ubiquitous robots has been implemented to validate the proposed approach.

In the next chapter, we are going to present the proposed framework in detail .

Chapter 4

Semantic Privacy Management Framework

Contents

4.1 Introduction

In this chapter, we are going to represent a semantic framework that integrates a meta-model and reasoning tools allowing any ubiquitous system designer to easily implement mechanisms to manage privacy policies.

In the first part of the chapter, we are going to give an overview of this framework. This latter is based on three layers : the meta-model level of privacy policies, the model level of privacy policies and the reasoning middleware for privacy management. Then, we are going to focus on the description of Model-Driven Engineering (MDE), mainly, the Ontology Definition Meta-model (ODM) and the Business Process Definition Meta-model (BPDM). Afterwards, we are going to introduce the semantic executable platform for mapping privacy policies. This platform is based on a reasoning based on ontologies and inference rules operating on the assumption of the closed world using the SmartRules language.

In the last part of the chapter, we show how privacy policy templates play a role in the expressiveness of the proposed meta-model. This latter is characterized by a high level of abstraction and expressiveness to define management policies of privacy regardless of the application domain and can be adapted to different contexts.

4.2 Semantic Framework Overview

As we have argued in the chapter 3, Web Ontology Language (OWL) is the most suited formal tool to build a common sense and a machine understandable representation of domain knowledge, while Model-Driven Engineering (MDE) is the appropriate approach for building well-designed systems. However, we use, usually, OWL to create a flat description of knowledge where instances, domain concepts (models) and upper concepts are represented in the same document or knowledge base. Some attempts exist to promote the use of web-based upper and domain ontologies, in general for knowledge retrieval purposes. These ontologies are composed, in general, of a huge number of concepts and properties ranging from hundreds to millions. Reusing these ontologies may pose several design and implementation issues for complex systems that involve reasoning modules with a medium or a small size of knowledge bases. In addition, OWL provides an executable environment for knowledge retrieval but its underlying reasoning based on open world assumption prevents its use for defining reactive control behaviors in complex systems running in dynamic domains. These later require a closed world assumption reasoning. Therefore, additional features are needed in OWL for overcoming this limitation.

With respect to MDE, the design of complex systems is easily undertaken with the UML graphical notation according to a modeling approach based on meta-models. MDE suggests, also, an efficient approach for the definition of new meta-models by the assembly of a well-designed modular blocks belonging to meta-models standardized by the OMG, thanks to the Meta Object Facility (MOF) representation standard. To overcome the main limitations of using the UML notation for the definition of ontology knowledge models, an attempt was the definition of a meta-model for OWL language, which is standardized by the OMG under the name Ontology Definition Meta-model (ODM).

In summary, there are differences in the way of intelligent control systems are designed according to the MDE approach or to the semantic web approach based on OWL. Therefore, the appropriate approach for handling the design of reactive control system such as privacy management system would be an hybrid framework that combines the benefits of both MDE and semantic web tools.

The main contribution of this thesis is an hybrid framework that provides modelling tools for simplifying privacy management in any architectural model of ambient intelligence and ubiquitous robotics applications. The proposed framework proposes a new meta-model for privacy policies. It suggests a privacy by design approach that follows all the stages of the MDE approach. Consequently, bad design choices are avoided in handling privacy, ranging from the Computational Independant Model (CIM) definition to the Platform Independant Model (PIM) implementation according to a standard middleware for applying privacy policies controls through enforcing or entitlement. Moreover, the clear separation of modelling concerns in MDE allows designers distinguishing between what is generic - related to the common sense view of privacy - and what is specific to the application. To adopt the same design principles, the OMG XMI standard format can be easily used to exchange the meta-model and derived models between designers of AmI and ubiquitous robotics applications that may interact together.

The architecture of our proposed framework is structured according to three layers corresponding to the following levels of privacy management modelling : meta-model level, model level and privacy control implementation level through a reasoning middleware platform, see figure 4.1.

4.2.1 Meta-model Level of Privacy Policies

This level provides a meta-model for the conceptual description of privacy policies. Indeed, this meta-model is not only a formal or a graphical representation of concepts that is valuable for getting more readable knowledge model for defining privacy policies, but it provides, also, a much more consistent approach, MDE compliant, for the common sense description of what must be the privacy management. The proposed meta-model simplifies the definition of privacy policies by proposing a restricted set of upper concepts. The proposed concepts have roots or are associated with the Ontology Definition Meta-model (ODM) and Business Process Definition Meta-Model (BPDM).

BPDM offers a set of modelling constructs to define business processes - activities or tasks - that involve actors belonging to the same or to different organizations and communities. While ODM provides the constructs needed for defining the concepts for access control based policies. Similar to the XACML notation, the BPDM

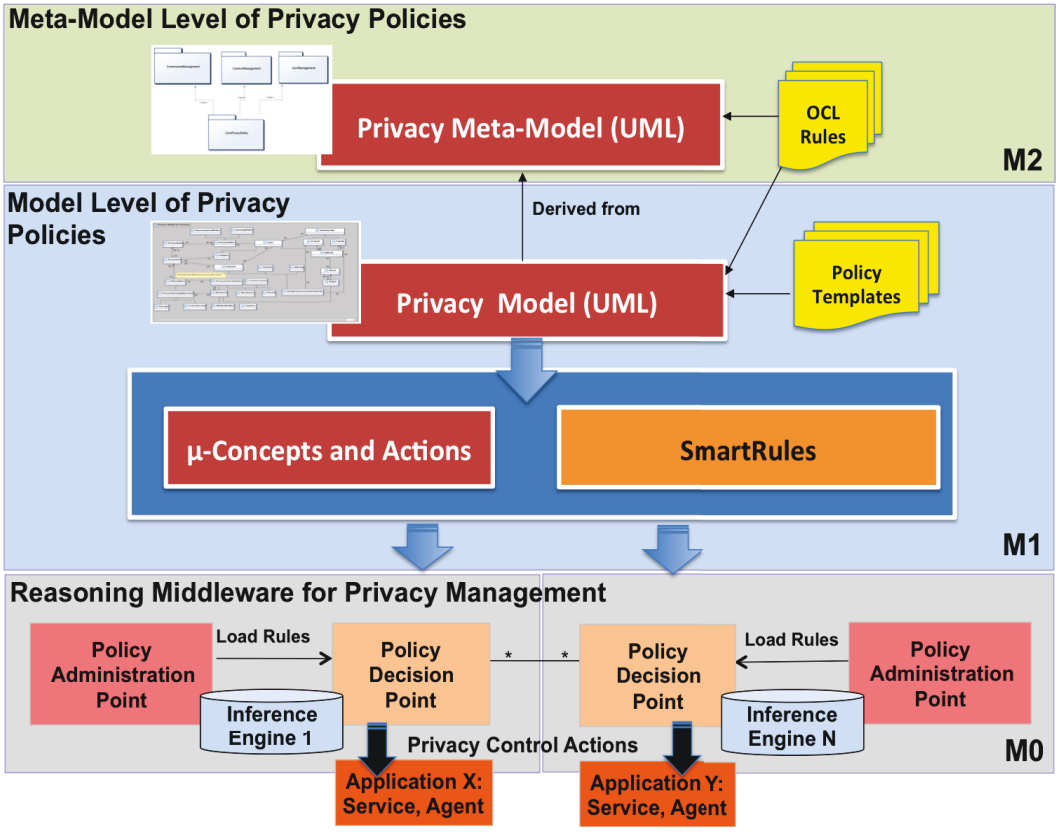

FIGURE 4.1: Privacy Meta-modelling Management Framework Overview.

provides the constructs needed to define privacy obligations as a set of sequential processes involving tasks that must be executed to enforce the privacy control during and after the access or use of sensitive personal resources. These tasks can be executed by any generic system for process orchestration or choreography. Both ODM and BPDM are based on Meta Object Facility (MOF) and UML profile. A detailed description of ODM and BPDM is given in section 4.2.4.1 and 4.2.4.2.

To the proposed meta-model, we associate generic n-ary templates of typical privacy policies and Object Constraint Language (OCL) constraints for defining and checking at the same time the validity of the defined privacy policies. We propose four templates of privacy policies that cover most to the privacy management use cases, namely : (i) Release Policies for controlling how the personal information will be disclosed owner and to whom (ii) Data Handling Policies for controlling the transmission, processing and storage of personal information by the third party (iii) Policies that can be applied on sensors for controlling their observations (iv) Policies for protecting the disclosure of privacy policies to the third party.

The Object Constraint Language (OCL) is used in this level - M2 level - not only to bring model-theoretic semantics of the class definitions such as class restrictions, qualified restrictions, etc, but, also, to define other restrictions that avoid the definition of wrong privacy policies. OCL rules can be used, also, at the levels M1 and M0 to validate additional constraints of the privacy control system architecture and implementation. However, OCL language has neither a formal model nor a formal proof theory. Therefore, we can make automated reasoning over the UML models, by mapping them to either production rules or description logic rules.

To avoid such drawback and keep the meta-model highly abstract, and in the same time usable easily, we will not make the mapping on the meta-model layer. This latter combines both UML and OWL modelling features and can be used to define production rules for controlling and monitoring systems behaviours.

4.2.2 Model Level of Privacy Policies

At this level, we make use the concepts of the meta-model and take into consideration the policy generic templates for specifying in UML privacy policies that will be applied for a particular application domain. The representation and reasoning on privacy policy models are done also under a Closed-World Assumption (CWA). Negation as failure is the main feature of this reasoning, which is important to define default controls when some policies are not matching.

To define such policies, we propose to prefix policy variables in the antecedent with a negation operator. In the case of a negative match with the instances stored in the knowledge base, the inference engine will trigger the actions that are defined in the consequence part of the policy. In an open world assumption, it will be assumed that if some statements are not explicitly asserted in the knowledge base, they remain possible and the reasoner decision will be "don't know". In this case, we cannot guarantee that the privacy control system will behave properly in the case where a request is not matching all the policies stated in the knowledge base.

CWA is, also, an important feature in the meta-model definition. UML and OCL support natively the closed world assumption. CWA avoids that we define properties of privacy concepts without specifying the range and the domain. When we make the mapping of these concepts from UML to OWL, which is based on the

open world assumption, distinct UML properties are interpreted as one property in OWL, which can lead to a form of inconsistent model. Restrictions can be added to the properties to prevent such drawbacks, which unfortunately adds another layer of complexity to the meta-model.

Considering that UML is not an executable model for knowledge management, the intuitive approach is to make a mapping of the policy model to an executable model, which supports closed world assumption like in databases and semantic description like in OWL. For this purpose, we propose, in our approach, to make a mapping of policy rules defined in the UML model to rules in the SmartRules language, while concepts that are used to define policy rules are mapped to concepts and actions in the µ-concept ontology language. This latter is a variant of OWL 2 that supports closed world and unique name assumptions.

4.2.3 Reasoning Middleware for Privacy Management

At this level, we propose a middleware that can be used by application designers to implement a reasoning module. This latter can make privacy management policies on behalf the application. This approach has similarity with XACML based access control system where the authorization and enforcement control decision are delegated to specific components, namely Policy Decision Point (PDP) and Policy Enforcement Point (PEP). The Policy Administration Point (PAP) is used for storing access control policy. PAP creates security policies and stores these policies in the appropriate repository. PDP evaluates the applicable policy and renders an authorization decision. In our approach the PDP and PEP are encapsulated in one entity "the PDP", which is an inference engine running production rules. The PAP stores the policy rules, expressed in the SmartRules language, which are mapped from the policy models.

4.2.4 Description of the Foundational Meta-models

4.2.4.1 Ontology Definition Meta-model

ODM includes five meta-models that could be extended to build conceptual description for a particular domain application. At the core of the ODM, the Description

Logic (DL) meta-model provides the concepts for defining the Tbox and Abox on an ontology in the UML notation. The Common Logic (CL) meta-model is a declarative first-order predicate language that is used for exchanging semantic knowledge over an open network according to the ISO specification. CL allows ontology designers - ontologists - using the ODM to be able to express constraints and rules with expressiveness beyond that supported by description logics. This is done through a variety of different syntactic forms, called dialects, all expressible within a common XML-based syntax and all sharing a single semantics.

The other three meta-models concern more structural or descriptive representations and they are : Resource Description Framework Schema (RDFS) [154], Web Ontology Language (OWL) [155] and Topic Maps (TM) [156]. Each of these meta-models within ODM is identified by an UML profile. Hence, we find, for example, an UML profile for RDF, OWL and TM. In addition, RDF meta-model generalizes OWL meta-model. This latter contains OWLBase, OWLDL and OWLFull meta-model concepts that inherit, respectively, from RDFS and RDFWeb meta-models.

The ODM specification offers several benefits to potential users [157] : (i) Many options in the level of expressivity, complexity, and form are available. They aim to facilitate the design and implementation of conceptual models, ranging from familiar UML and ER methodologies to formal ontologies represented in description logics or first order logic. (ii) The use of standards languages and tools, grounding in formal logic, and model-theoretic semantics for the knowledge representation languages supported enable reasoning engines to understand, validate, and apply ontologies developed using the ODM. (iii) With profiles and mappings techniques, exchanging models, developed independently in various formalisms, becomes easy and quite sufficient. In addition, they enable consistency checking and validation. (iv) Marring MDA and semantic web technologies to support semantic web services, ontology and policy-based communications and interoperability, and declarative, policy-based applications in general is, also, the goal of ODM.

4.2.4.2 Business Process Definition Meta-model

A graphical representation of a business process model is important to define neutral description of process with regards to its implementation in target systems. The Process Definition Metamodel (BPDM) provides conceptual constructs based

on the BPMN notation, which are sufficient to represent most normal forms of interoperable business process and makes the specification of business process models an easy task. Moreover, BPDM provides an integrated and a consistent support for defining the semantics of all BPMN notation concepts, thanks to the MOF representation.

Process models can be seen as a service composition that can be executed through coherent models of "Orchestration" and "Choreography". These latter are considered as the most prominent approach of implementing complex assistive systems on web services for AmI or ubiquitous robotics. In addition, BPDM provides constructs to specify, also, performance, enactment, and execution of processes to take into account Quality of Service or obligations policies. An event-oriented approach is used to define process-monitoring events, such as the starting, ending and aborting the execution of processes and the way in which the execution can be sequenced in time.

While BPDM allows design process models that are easy to understand by designers it remains complex for users involved or concerned by these processes. A declarative language would be more helpful to allow the specification or mapping of processes from UML notation into a set of business rules that can be easily understood by both end users and designers.

4.2.4.3 Semantic Executable Platform for Mapping Privacy Policies

In this section, we represent the SmartRules language and the µ-concept ontology language that provide jointly an executable platform of privacy policy models. These languages were developed by Thales and LISSI lab in the context of the SembySem ITEA2 project and are provided with rule authoring and execution environment.

The SmartRules language allows to set up reactive rules for monitoring systems following the standard format of production rules (if "condition" then "consequence") [158] where we make only use of highly conceptual entities called µ–concepts, actions, and properties. Unlike SWRL, the SmartRules is based on intensive use of variables that can be bound to µ-concepts or properties. These latter allow

addressing any use case that needs reactive reasoning following the closed world assumption.

The µ-concept language is used to define, in micro ontologies, all the conceptual entities that are needed for SmartRules authoring in an application domain. These entities are represented in a similar way as in the OWL/RDF ontologies. The main differences with OWL is that µ-concepts language is dedicated for modelling real world objects by using individually or composing together a set of µ-concepts according to the unique name assumption, which means that each instance of a µ-concept can be associated with a formally identifiable real-world object. Real-world objects can be physical or immaterial entities.

Two different instances cannot refer to the same object and the name of the instance must be unique to avoid any possible contradiction. The property value of an instance can be declared in the µ-concept native representation as a literal inside the instance declaration as well as it can be declared as a constant. In figures 4.2 and 4.3, we denote the representation of µ-concepts and their properties that are used for rule authoring. The multilingual support is another differencing point with OWL that makes µ-concept ontologies, and respectively SmartRules, readable in several natural languages. For instance, a set of attributes are given to define a label and a description in a particular language for each element of the model. In addition, the figures 4.2 and 4.3 show an example of privacy policy concept and its properties declared respectively in RDF and the µ-concept native representation.

```
<smc:Concept rdf:ID   = "ReleasePolicy">
<rdfs:subClassOf rdf:resource   = "#PrivacyPolicy">
<smc:restriction>
  <smc:PropertyRestriction>
  <smc:onProperty rdf:resource = "#Time"/>
  <smc:default rdf:datatype="&xsd;integer">10</smc:default>
  </smc:PropertyRestriction>
</smc:restriction>
</smc:Concept>
```

FIGURE 4.2: Release Policy µ-concept with Property in RDF Format.

An instance I is associated to a concrete manageable object that is described using a single concept. Any number of P values could be associated to the same property. Two different instances cannot refer to the same object (the name of the instance must be unique to avoid any possible contradiction). The property value of a concept instance can be declared in the µ-concept native representation as a literal

```
<smc:Concept smc:ID = "ReleasePolicy">
  <smc:Inherits smc:Reference = "PrivacyPolicy"/>
</smc:Concept>
<smc:Property smc:ID = "Time">
  <smc:Default smc:LiteralType = "Integer">10</smc:Default>
</smc:Property>
```

FIGURE 4.3: Release Policy µ-concept with Property in SMC Format.

inside the instance declaration as well as it can be declared as a constant, see figures 4.4 and 4.5.

```
<smc:Instance smc:ID = "PrivacyPolicy1" smc:Concept = "PrivacyPolicy">
  <smc:Description xml:lang = "en">Alice Privacy Policy</smc:Description>
  <smc:PropertyValue smc:Property = "UserName" smc:LiteralType =
"String">Alice</smc:PropertyValue>
</smc:Instance>
```

FIGURE 4.4: µ-concept Instance in RDF Format.

```
<smc:Instance smc:ID = "PrivacyPolicy1" smc:Concept ="PrivacyPolicy">
  <smc:Description xml:lang = "en"> Alice Privacy Policy </smc:Description>
  <smc:PropertyValue smc:Property = "UserName" smc:Constant ="true"
smc:LiteralType = "String">Alice</smc:PropertyValue>
</smc:Instance>
```

FIGURE 4.5: µ-concept Instance in SMC Format.

The main differences with OWL concern that every µ-concept is used to describe the aim of manageable objects and can then define "actions" that each instance (or concrete manageable object) depending on this µ-concept will be able to execute [159], [160]. The semantic description of an action concept differs from a µ-concept only in the way that inverse-functional and inverse properties must not be used to describe an action concept.

The action concept description may include, also, restrictions on properties that are separated in two parts. The first one concerns properties specifying the restrictions when the action is launched (input restrictions) while the second one is related to the restrictions when the action is complete (output and effect restrictions). These restrictions behave exactly like property restrictions declared in µ-concept, overriding the default property behaviour on its domain, see figure 4.6.

The condition pattern of SmartRule is characterized by a priority level and the possibility to put single or a set of constraints on variables that are bound to µ–concepts or properties. A rule expressed without restrictions allows us to match

```
<smc:Action rdf:ID = "disclose">
  <smc:actionDomain rdf:resource="#ReleasePolicy"/>
  <rdfs:Label xml:lang="fr">divulguer</rdfs:Label>
  <rdfs:Label xml:lang="en">disclose</rdfs:Label>

  <smc:Property rdf:ID = "PrivacyPolicy" smc:Input="true">
    <MinCardinality>1</MinCardinality>
  </smc:Property>

  <smc:Property rdf:ID = "Message">
    </rdfs:domain rdf:resource = "#disclose">
    </rdfs:range rdf:dataType = "&xsd;string">
  </smc:Property>
</smc:Action>
```

FIGURE 4.6: Action Description in µ-concept Ontology Language.

every instance of a given concept. The consequent part of SmartRule is a declaration of the actions to be performed on µ-concepts instances corresponding to real world object.

The term action is a core language construct of the µ-concept language that does not exist in OWL. The semantic description of an action differs from a µ-concept - and owl class - in the way that inverse-functional and inverse properties must not be used to describe an action concept. We denote the following actions types : create or remove instances in/from the knowledge base, update property values or global variables, while the most important among them corresponds to a "do" action on an instance. Actions should be modeled always by considering the semantic description of the corresponding µ-concepts concerned by the action that will be executed [159], [160].

For this purpose, the action concept is characterized by a set of special properties describing the agent (actuator), the object of the action, the source of the command, the beneficiary from the execution, the modality, the topic and the context in which the action is or may be executed [160],[159]. All these properties are optional, with the exception of 'ActionAgent', whose presence is mandatory. It allows the SembySem execution environment to make easily the mapping between an action, instantiated from the conceptual level, with the actuator existing in the real world.

Apart from the possibility of defining and executing complex actions on the real world objects, other features of the µ-concept and SmartRules Languages that are missing in the W3C languages concern : (i) the possibility of defining inequality

constraints on the properties and, properties calculated as a combination of other ones and (ii) the modeling of actions and their corresponding concepts is less constrained compared to the 'binary' way imposed by OWL or DL to describe relations among µ-concepts et actions. For instance, actions can be associated with "Multivalued (multi-cardinality) properties" and the use of the keyword "one" in SmartRule antecedent – only with multivalued properties – in order to select one value from all those corresponding to a multi instances matching can then be considered as another way of getting rid of the limitations associated with SWRL and OWL languages. We can consider that the µ-concept language offers a modelling closer to the Entity/Relationship and UML style [159]. For instance, the following rule in the figure 4.7, allows the triggering of an event each time a sensor observation is done in a private area.

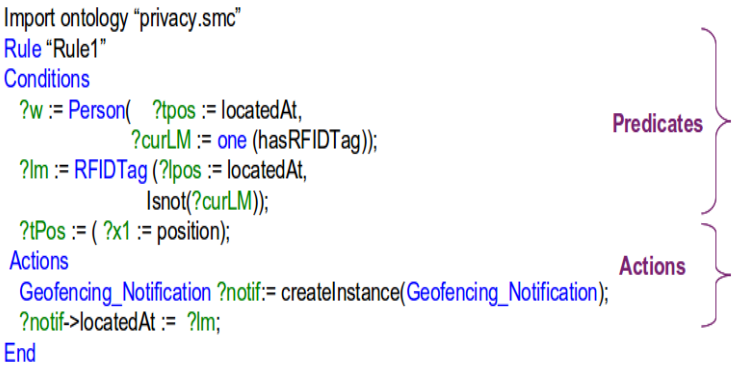

FIGURE 4.7: RFID based Position Tracking in the SmarRules Language.

4.3 Privacy Meta-model for Ambient Intelligence

In this part of thesis, we describe our contribution (mainly the privacy meta-model) to the field of privacy in AmI environment. First, we detail the privacy policy templates. Then, we point out each meta-model component and concept. Finally, we show the corresponding OCL constraints and rules.

4.3.1 Privacy Policy Templates

Privacy Policies can be defined according to specific templates. Privacy rule is prefixed by the policy template name and composed of two parts : antecedent

and consequent. The meta-model provides modelling constructs to define policy according to the general form antecedent then consequent, which can be read informally as : If a Privacy Sensitive Operation occurs on a private ressource and context holds then apply Privacy Control decision that can be reactive atomic action or process including obligations. Obligation can be considered, also, as a rule, which can be read if Privacy Control decision then Execute obligation process. The following statements, as depicted in the figure 4.8, form a privacy policy template. They provide a kind of grammar that we have used to design our meta-model for policies definition. Consequently, a privacy rule template has the following canonical form :

privacy rule ::= 'Policy Template ('**policy-template**' Implies ('**antecedent consequent**'))'
antecedent ::= 'Antecedent (' [Not |One] **privacy-sensitive-operation context private-resource** ')
consequent ::= 'Consequent ('**privacy-decision**')'
policy-template ::= [access control policy |release policy |data handling policy |sensing policy |policy disclosure policy]
privacy-sensitive-operation ::= [observation |transmission |processing |modification |storage]
context ::= [time [location [situation [activity]]]]
privacy-conrol-action ::= [authorization |prohibition |obfuscation |anonymization |generalization |disclosure |distruption |restriction |disable |enable]

privacy-resource ::= [uri |sensor |actuator |physical object |immaterial object]
privacy-decision ::= [privacy control action |privacy obligation process]

FIGURE 4.8: Privacy Rule Template.

As depicted in the figure 4.8, privacy sensitive operations concern the operations of Observation, Transmission, Processing, Modification and Storage. For the observation, the user just simply read personal information or resource in a very specific context. To do this, the necessary actions that are related to the observation are permit (or allow) and deny (or prohibit) the observation.

The transmission of sensitive and personal data is considered as the most critical and delicate operation in the protection of user privacy. Any information sent

to individuals, information systems or places at a website can be used in any circumstance including but not limited to dissemination, reproduction, transmission, publication, diffusion and placing of content on the web. For example, where an email is used to transmit this information, security mechanisms are required for standard email. This can lead to serious abuse, because it opens the way for the data to be used for purposes quite different from its intended use. This can happen for a number of reasons. The rules governing who can use the data and for what purpose may not be clear or restrictive enough to protect the intentions and interests of the subjects. Or those who control the data may not enforce the rules.

The operation of processing concern in general the use of personal information as input for any processing. A typical example is using the list of person's contacts in a mailing system. Modification is a kind of processing users that can grant to the third party the restricted right to reveal private information but it can oblige the third party to make a transformation of these data (such as transforming the age to an age category).

It is obvious that the users have no longer physically possess the storage of their data. However, by the operation of "storage", we mean that the user could control the storage of his personal data such as medical data or location data,in the system. In other words, the data holders themselves are responsible that the released information does not affect privacy by permitting or denying the storage of the personal information.

Through the following privacy control actions, our purpose is to enhance the integrity of the transmitted information by an authorization, anonymization, obfuscation, disclosure and restriction actions.

Authorization action. Individual could deny or permit the access to his personal data or to his ressource.

Anonymization. In privacy preserving data publishing, in order to prevent privacy attacks, data should be anonymized properly before it is released. Anonymization corresponds to a technology that converts clear text data into a non human readable and irreversible form. This latter is not limited to preimage resistant hashes (e.g. one-way hashes) and encryption techniques in which the decryption key has been discarded. Indeed, data is considered anonymized even when conjoined with pointer or pedigree values that direct the user to the originating system, record,

and value (e.g., supporting selective revelation) and when anonymized records can be associated, matched, and/or conjoined with other anonymized records. In our thesis, we just consider the anonymization in the AmI environments and users' interactions. In other words, data anonymization enables the transfer of information across a boundary, such as between two agents or to users while reducing the risk of unintended disclosure in a manner that enables evaluation and analytics post-anonymization. We note that there are several K-anonymization algorithm proposals in the literature. Anonymization methods should take into account the privacy models of the data and the utility of the data. Generalization and perturbation are the two popular anonymization approaches for personal data.

- ***Generalization*** is used for privacy-preserving data collection but has not been used for privacy-preserving data collection. There are several probabilistic privacy measures based on a distribution attack and use it to define the respondent's problem of finding an optimal anonymous data.
- ***Perturbation*** uses randomized techniques to mask the data for preserving the privacy of sensitive data. This methotodology attempts to hide the sensitive data by randomly modifying the data values often using additive noise.

Obfuscation. Generally, for privacy purposes and preservation, obfuscation aims at making user personal data harder to understand or read. Data obfuscation is a form of data masking where data is purposely scrambled to prevent unauthorized access to sensitive materials. This form of encryption results in unintelligible or confusing data. Data Obfuscation techniques distort data in order to hide information. Many data obfuscation techniques have been suggested and implemented for privacy preserving data applications. Individuals using obfuscation should be able to balance their desired level of privacy against their desired quality of personal data such as its location, name, age, etc.

Disclosure. In our thesis, we focus on self-disclosure which is the process of communicating information about the self to another person. Indeed, personal information are shared with users or revelated to systems. Moreover, we focus on the concept of minimal disclosure that defines the disclosure of personal data to third parties shall be restricted and only occur upon certain conditions.

Restriction. A privacy statement specifies restrictions that have to be satisfied before that the access to personal data is granted. If just one condition is not satisfied, the access should not be granted.

4.3.2 Privacy Meta-model and the MOF

As depicted in the figure 4.9, this privacy meta-model is conformed to the MOF layered architecture.The meta-model appears at the M2 layer and inherits from ODM and BDM meta-models at the same time. It exists with the standardized meta-models such as UML meta-model and, also, with any proposed meta-models. The table 4.1 shows the positioning of our meta-model in the MOF architecture with some examples. The M1 layer as it is an instantiation of M2 layer, the results of our privacy meta-model are privacy models that depend on the system designer's choices and his technical orientations. In the next section, we are going to detail this privacy meta-model and highlight its different components.

Meta-level	MOF terms	Examples
M3	meta-metamodel	MOF model
M2	meta-model, meta-meta-data	privacy meta-model
M1	model meta-data	UML models privacy models
M0	object data	modeled systems ubiquitous modeled data

TABLE 4.1: OMG Meta-data Architecture and Privacy Meta-model.

4.3.3 Privacy Meta-model Specification

The privacy meta-model uses a hierarchical package structure to control the complexity, promote the understanding of its concepts, and support its reuse. The meta-model concepts are contained in a privacy meta-model package as depicted in the figure 4.10. This latter represents the parent package that includes four other packages that will be described in the following subsections. We underline that the "CorePrivacyPolicy" package uses the other packages through the stereotyped association "use" as some concepts depends on the other ones included in community, context and user management. The figure 4.10 depicts this core privacy policy and the other packages.

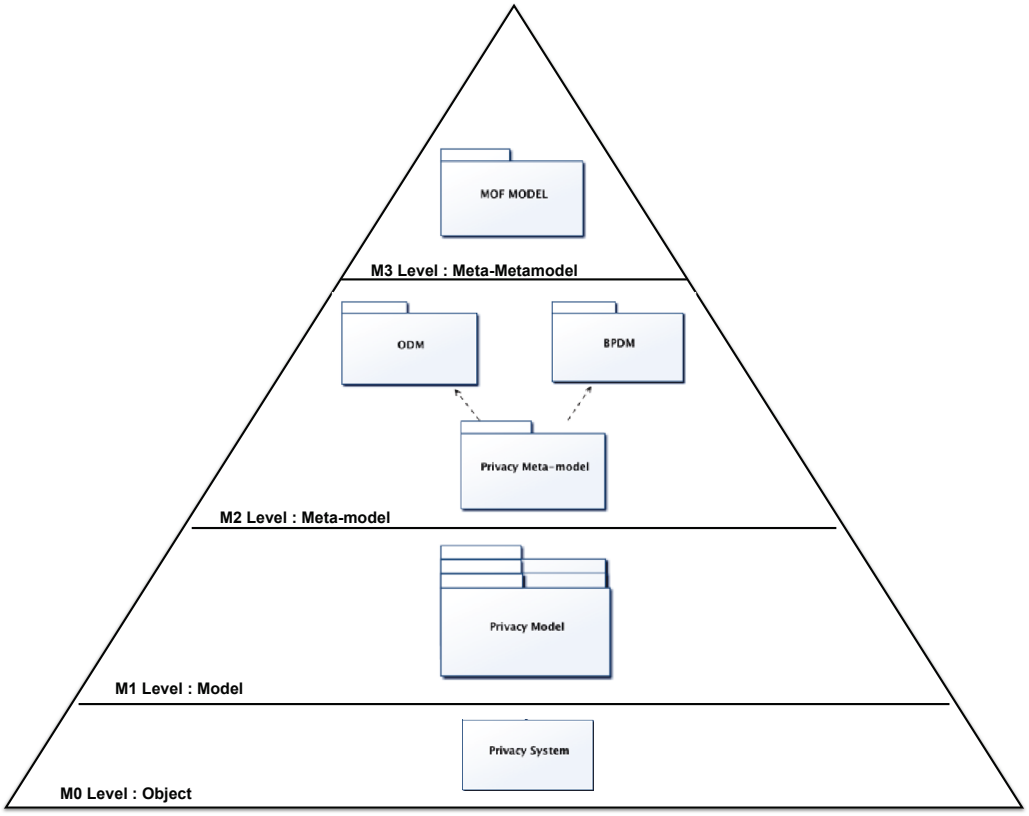

FIGURE 4.9: Privacy Meta-model and the MOF.

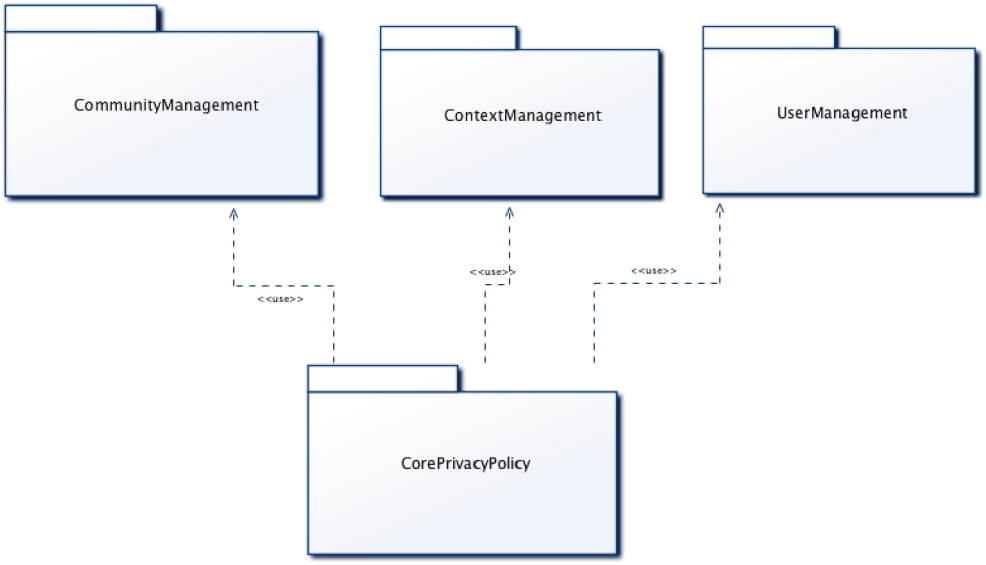

FIGURE 4.10: Privacy Meta-Model Organisation.

4.3.3.1 Privacy Policy Core Concepts

Every concept of the CorePrivacyPolicy meta-model (represented through an UML class) inherits from the "Ontology" class through the ODM meta-model. CorePrivacyPolicy package includes six concepts : (i) PrivacyPolicy, (ii) PrivacyRule, (iii) Parameter, (iv) PolicyObject, (v) Action and (vi) Condition as depicted in the figure 4.11. The PrivacyPolicy class has as attributes a policy name and a date of creation. This privacy policy could be, a Release Policy, an Access Control Policy, a Data Handling Policy, a Sensing Policy or a Policy Disclosure Policy.

Privacy rule could be shown as a simple or a complex rule. This latter is considered as a rule flow or as a process as it inherits from the Process class of BPDM meta-model. A rule flow is a kind of succession between rules within a rule process. A Control Flow is connected to at least one rule, either as a triggered control flow or a triggering control flow. A rule flow is a graphical description of a sequence of steps that the rule engine needs to take, where the order is important. The rule flow can also deal with conditional branching, parallelism, synchronization, etc.

PrivacyRule class has as attributes a rule number and a rule name. A simple privacy policy rule has at least three parameters. It is mandatory to have as a first parameter an owner or an initiator of the rule, as a second parameter a receiver of the rule, a third parameter a policy object and context and situation as additional parameters .

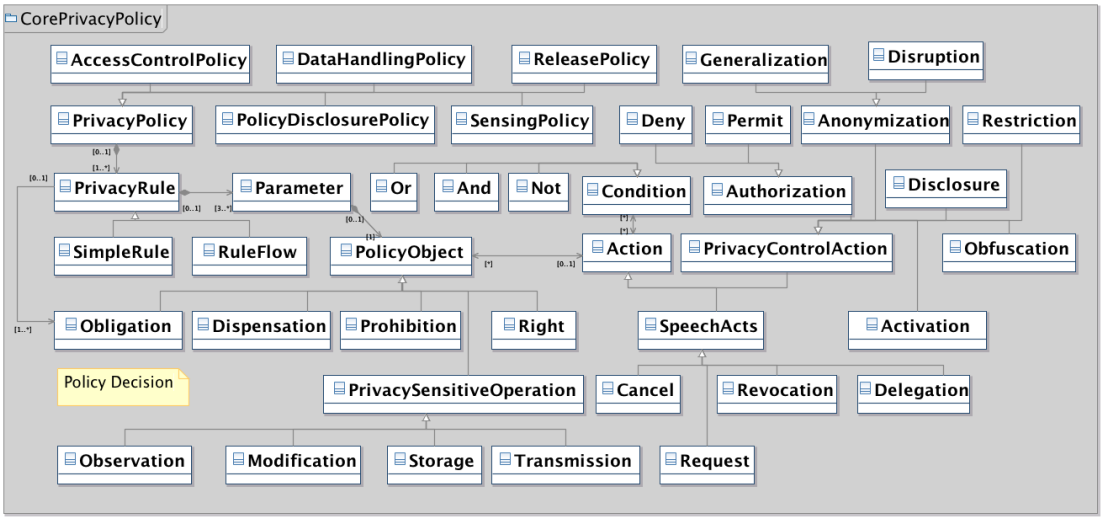

FIGURE 4.11: Core Privacy Policy Meta-model.

Parameter is used to define the components of the privacy rule. It is composed of a "Subject" (from UserManagement), a "PolicyObject", "Situation" and "Context" classes (from ContextManagement). It has as attribute the parameter number.

Policy object denotes the constructs of deontic concepts. "PolicyObject" is an abstract class that generalizes "Right", "Prohibition", "Obligation" and "Dispensation". In addition, it is based on privacy sensitive operations. Rights are permissions that a subject or an entity has. Prohibitions are negative authorizations implying that an entity cannot perform the action. Obligations are actions that a subject has to perform and are usually trigged when a set of conditions are true. Dispensations are actions that a subject is no longer required to perform. They act as waives for existing obligations. Moreover, policy object is shown as a privacy sensitive operation declined in observation, transmission, processing, modification and storage.

Action concept can be a privacy control action or speech acts. Privacy control action specifies the actions of anonymization, obfuscation, disclosure, restriction, authorization and activation/deactivation. The second class defines, particularly, four speech acts that affect the policy objects of the communicating entities : delegation, request, cancel, and revocation. These speech acts are also governed by policies and entities can only use a certain speech act if they have the right to it.

Condition concept is an abstract class that supports used to define logical operators in the antecedent of privacy rules.

4.3.3.2 Community Management

Every class of the CommunityManagement meta-model inherits, also, from the "Ontology" class as depicted through the stereotype $<< fromODM >>$. Community Management includes five concepts : (i) Subject, (ii) Community, (iii) Resource, (iv) Set and (v) Operator as depicted in the figure 4.12.

A subject represents any entity related to a domain. "Subject" class is abstract and could be a user, a role or a group of users. A community could be an organization, a company or a government. It could be by one or many privacy policies related to a subject. A resource depicts any community resource that could be physical, an agent, software or any ubiquitous device but, also, a private resource or could

be an immaterial object. The concept "SET" includes many subjects with several operators. A set has at least two subjects. An operator points out the link between the subject or communities such as Union, Intersect or Minus.

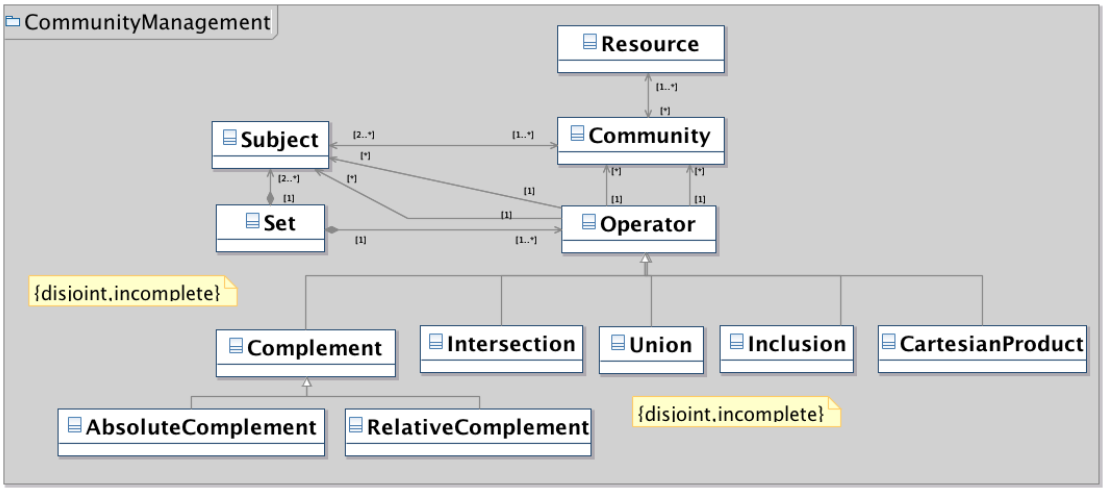

FIGURE 4.12: Community Management.

4.3.3.3 User Management

This package describes classes or concepts related to any entity that inherits from the "Subject" class. Every class inherits from the "Ontology" class. This is depicted through the stereotype $<< fromODM >>$. UserManagement includes three constructs : (i) Role, (ii) User and (iii) Group as depicted in the figure 4.13.

A role represents any job function in a community, an organization or a domain. As depicted in our meta-model, a user can play a role and could belong to many groups. The user could be also the owner of the privacy policy (generally an individual) or a third party. In fact, an individual may require assistance from a third party. Hence, it is possible that he/she authorises a third party to act on his or her behalf. Consequently, the third party has access to personal information with the consent of the individual.

4.3.3.4 Context Management

As usual, every class inherits from the "Ontology" class. This relationship is depicted through the stereotype $<< fromODM >>$. Context management

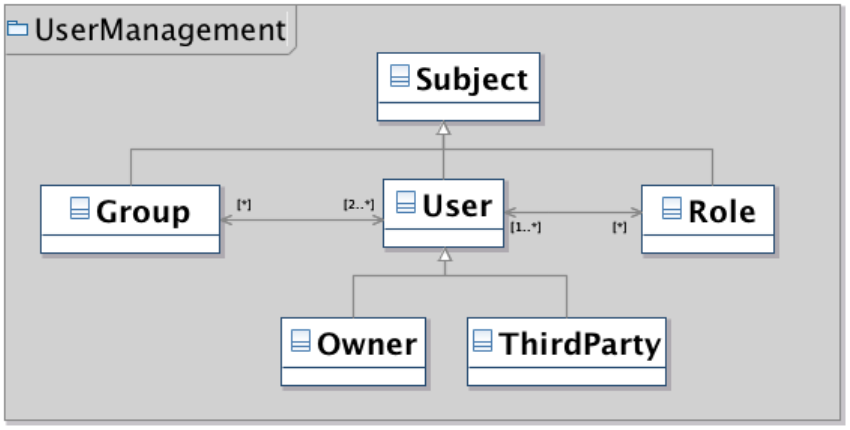

FIGURE 4.13: User Management.

includes two concepts : (i) Context (ii) and Situation as depicted in the figure 4.14. User context is an important element in our privacy meta-model as it forms a dynamic character that may influence the privacy policy. The physical context class defines the location particularly the position, orientation, velocity and trajectory. The situation denotes a set of context parameters that includes context process in a time period. In our meta-model, we devide the context into seven sub-concepts. Hence, "Context" class generalizes seven other classes and they are : Identity, Activity, Time, Emotion, Experience, Environment, ComputerEntityContext and Physical Context.

The activity class is the most important one. It denotes the activity or the task done by a user such as meeting, reading, working, walking, sleeping and sitting. The time class represents a sequence of events, duration while the emotion class describes the user's state of mind. The environment class shows the temperature, humidity, brightness and loudness. The computer entity context denotes the context of any computer entity such sensor, device and appliances. Finally, the physical context class defines the location particularly the position, orientation, velocity and trajectory. The situation denotes a set of context parameters that includes context process in a time period.

4.3.4 Privacy Meta-model Overview

In this part, we focus on the relationships and associations between privacy meta-model concepts of the different packages. The figure 4.15 depicts the whole privacy

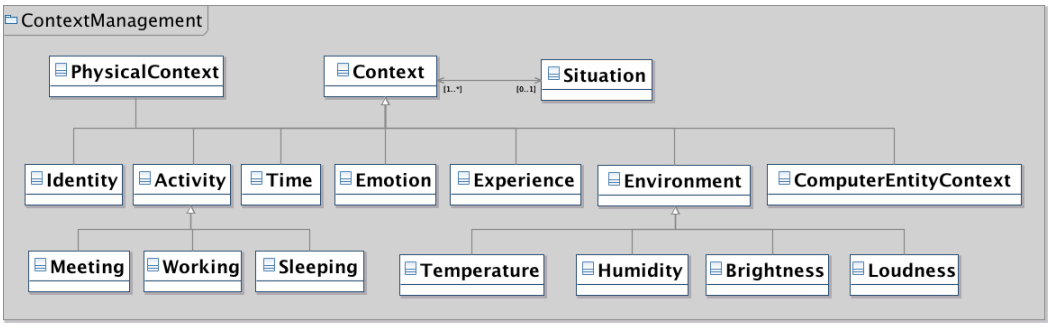

FIGURE 4.14: Context Management.

meta-model diagram that includes the four packages that we have already described previously and they are : Core Privacy Policy, Community Management, User Management and Context Management.

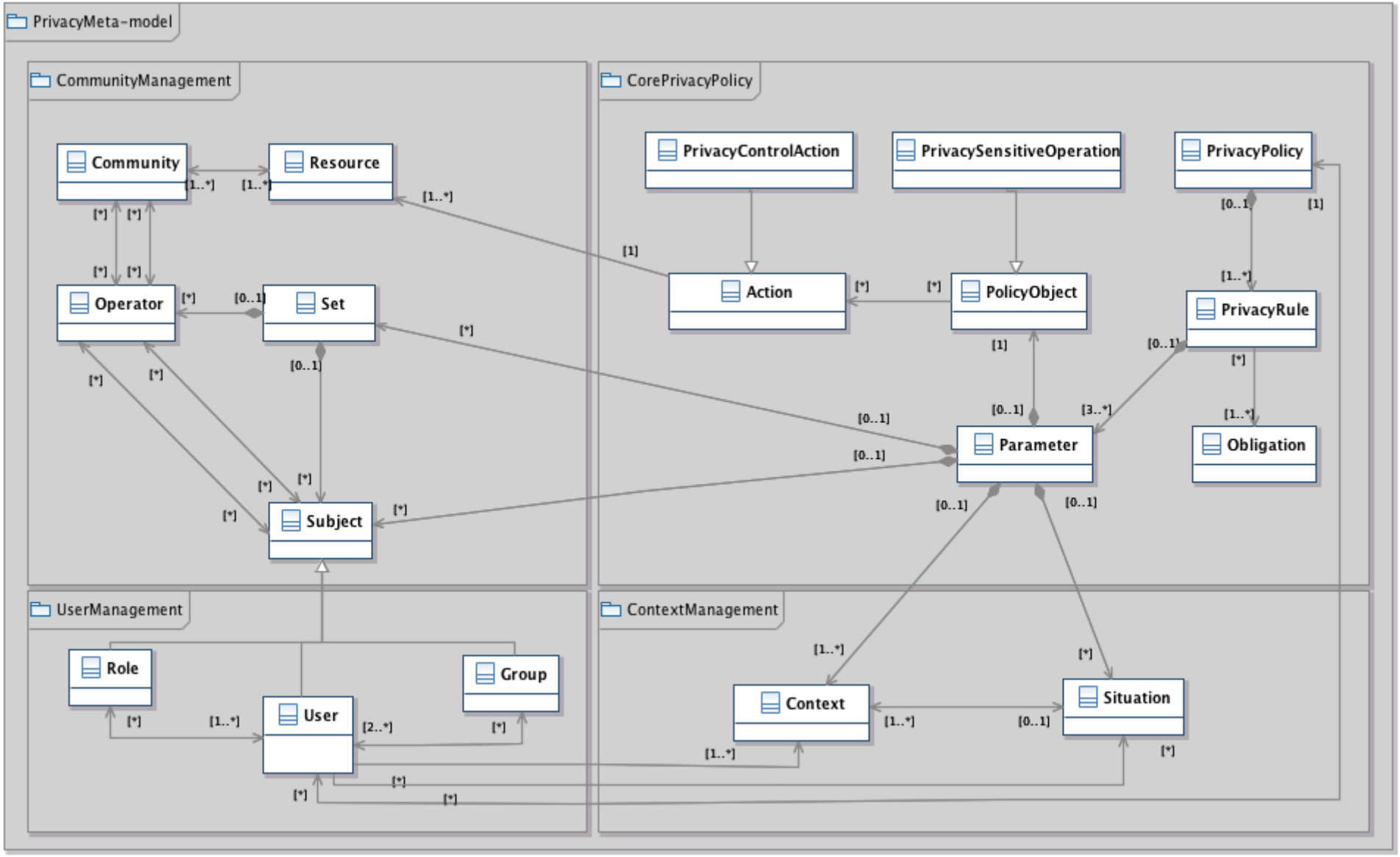

FIGURE 4.15: Privacy Meta-model Overview.

A user belongs to a community and has a privacy policy. This latter is composed by at least one privacy rule. This relationship is depicted in the diagram through the composition association and association occurrence "*". The parameter concept depicts five composition associations with other classes : "Set", "Subject", "Policy-Object", "Situation" and "Context". Hence, a parameter is formed by maximum a set of subjects, one subject, one policy object, of perhaps one situation and of at least one context. Otherwise, the privacy control action is applied on a private resource, in a specific time for a specific subject.

4.3.5 OCL Constraints Rules

Basically, UML diagrams are not enough. We need a language to help with the specification capabilities and that has to be an oriented object (OO) one. We opt for the Object Constraint Language (OCL) as it is used to specify constraints on OO systems.

OCL is not the only one available solution, certainly, but is the only one that is standardized. OCL is a formal language based on the first order predicate logic to annotate the UML diagrams by allowing the expression of such constraints. These constraints are used especially to describe the semantics of UML and its various extensions, participating in the definition of profiles. The privacy meta-model is supplemented by OCL invariants which serve three purposes : (1) They represent privacy-aware constraints, (2) check for reasonable policy designs, and (3) regulate the snapshot concepts. We aim to overcome the limits of UML and to complete our privacy meta-model by adding the necessary constraints.

For the privacy rule constraints, we define one subject or a set of subjects but not both of them at the same time [Set or Subject]. The third parameter is the policy object. We have only one policy object. The situation and context parameter are considered as the forth and fifth parameters. The first and second parameters form the privacy policy rules. Hence, they are compulsory as an exclusive choice (Subject XOR Set of Subject). The third parameter which is the policy object is compulsory. The fourth parameter which is context is compulsory while the fifth parameter which is the situation is optional.

However, as represented in the class diagram, we cannot express that we have, for the same privacy rules, compulsory and optional parameter and the exclusive choice between the first and the second one. But, these constraints could be fulfilled by the OCL language as depicted in the following rule of the figure 4.16 :

Context : Parameter
Invariant params : receiver.Set XOR receiver.Subject AND SecondParam.PolicyObject AND ThirdParam.ContextParam OR Fourth.SituationParam

FIGURE 4.16: Rule 1 : OCL Rule for PrivacyRule Class.

Regarding the community constraints, a community has at least one privacy policy. It groups at least two subjects and it includes at least one resource. This constraint is expressed in the figure 4.17.

Context : Community
Invariant : self.contains $->$ size $=$ 1 AND self.subjects $->$ size $=$ 2 AND self.resources $->$ size $=$ 1

FIGURE 4.17: Rule 2 : OCL Rule for Community Class.

Regarding the set constraints, a set is composed of at least one set operators and of at least two subjects as shown in the figure 4.18.

Context : Set
Invariant : self.setOperators $->$ size $=$ 1 AND self.subjects $->$ size $=$ 2

FIGURE 4.18: Rule 3 : OCL Rule for Set Class.

All the subclasses of Operator class are disjoint because the intersection of their sets is empty. Hence, we must enter the term "disjoint" in the specialization relation on the line linking the parent and child classes. Instead, we use the circle inheritance both hemispheres meaning empty disjoint and incomplete.

Otherwise, Operator class has as a left operator, a community or a subject (one of them at least or the both). It has as a right operator a community or a subject (one of them at least or the both). But if we have one left operator as a community,

Context : Operator
Invariant : s : if (s.isKindOf(Intersection) OR s.isKindOf(Union) OR s.isKindOf(Inclusion) OR s.isKindOf(CartesianProduct)) then
((C.RightOP XOR S.RightOP) AND (C.LEFTOP XOR S. LEFTOP))
EndIf

FIGURE 4.19: Rule 4: OCL Rule 1 on Operator Class.

Context : Operator
Invariant : s : (s.isKindOf(Intersection) XOR s.isKindOf(Union) XOR s.isKindOf(Inclusion) XOR s.isKindOf(Complement) XOR s.isKindOf(CartesianProduct))

FIGURE 4.20: Rule 5: OCL Rule 2 on Operator Class.

we cannot have a subject as a left operator as shown in the figures 4.19 and 4.20 .

Otherwise, the user class must have one context at least or one or more situation as depicted in the figure 4.21.

Context : User
Invariant : self.hasContext $->$ size $>= 1$ or self.hasSituation $->$ size $>= 0$

FIGURE 4.21: Rule 6: OCL on User Class.

The Role, User and Group classes are disjoint because the intersection of the sets of roles, users and groups is empty. ($\forall r \in Role, \forall u \in User and \forall g \in Group, r <> u <> g$). Hence, we must enter the term "disjoint" in the specialization relation on the line linking the parent and child classes as shown in figure 4.22.

Context : Subject
Invariant : s : (s.isKindOf(Role) XOR s.isKindOf(User) XOR s.isKindOf(Group))

FIGURE 4.22: Rule 6: OCL on Subject Class.

4.4 Conclusion

In this chapter, we have motivated the need for a framework of privacy by design of AmI applications. The framework allows for defining privacy policies by using a policy template according to a meta-model proposed as the main contribution of this thesis. We have explained the role of the main concepts of the meta-model and why we have reused the two well known meta-models ODM and BPDM in the modelling of privacy policies.

Our framework takes into account the main requirements for privacy management by adopting Model-Driven Engineering (MDE), common sense knowledge description with ontologies and privacy policy control through reasoning according the closed world assumption. Defining privacy policies according to a meta-model and generic template allows for an interoperable and coherent control for the protection of user's privacy over heterogeneous third party domains. The templates are used for implementing policies as inference rules in the SmartRules language. The latter are defined with the concepts of the meta-model and run on forward chaining inference engines according the closed world assumption.Unlike some approaches that are based on the SWRL inference language, the SmartRules language allows for using variables on concepts, which are mandatory for defining adaptive policies and rules with negation. These features are not supported in SWRL.

The proposed framework incorporates, also, a generic middleware architecture for implementing and monitoring the execution of privacy policies.

In the next chapter, we are going to present the implementation of a proof of concept to validate the proposed framework according to an MDE methodology.

Chapter 5

Design and Implementation

Contents

5.1 Introduction

In this chapter, we will explore how to implement our proposed framework described in Chapter 4 in practice. We, first, proceed by following the MDA development processes that we introduce in the first section of the chapter. In fact, we aim at showing the simplicity and the guidance our meta-model in thanks to the MDA technologies. Thereafter, we will introduce the Human-Robot Interaction scenario through which we test our framework. We will justify the choice of the privacy policies and the concepts used in the application domain of the scenario. Then, we will show the usefulness of the privacy rules expressed in the SmartRules language. We focus and distinguish between normal situations of daily living and the emergency situation. Finally, we are going to describe the LISSI laboratory ubiquitous platform which is resulting from the European project SEMBYSEM.

5.2 MDA Application in the Software Development

Applying MDA to a software development follows a development process that is proposed by Mike Rosen as shown in the figure 5.1. The development cycle starts by modelling a business model. This model results from the analysis phase. It is, also, known as CIM (Computation Independent model). Then, from the CIM, we add the necessary information for the PIM (Platform Independent Model). Consequently, we make the transformation from the model to get the PSM (Platform Dependent Model) from a PIM. The process of model transformation can be performed automatically or manually. Finally, since a PSM, it can be the generation of source code to get the source code that is ready to deploy. All of these models represent from a design perspective, organizations models that capture different views of them.

5.2.1 Computation Independent Model

The strategy of deriving software requirements from business organizations is the aim of MDA based software development [161]. This software development

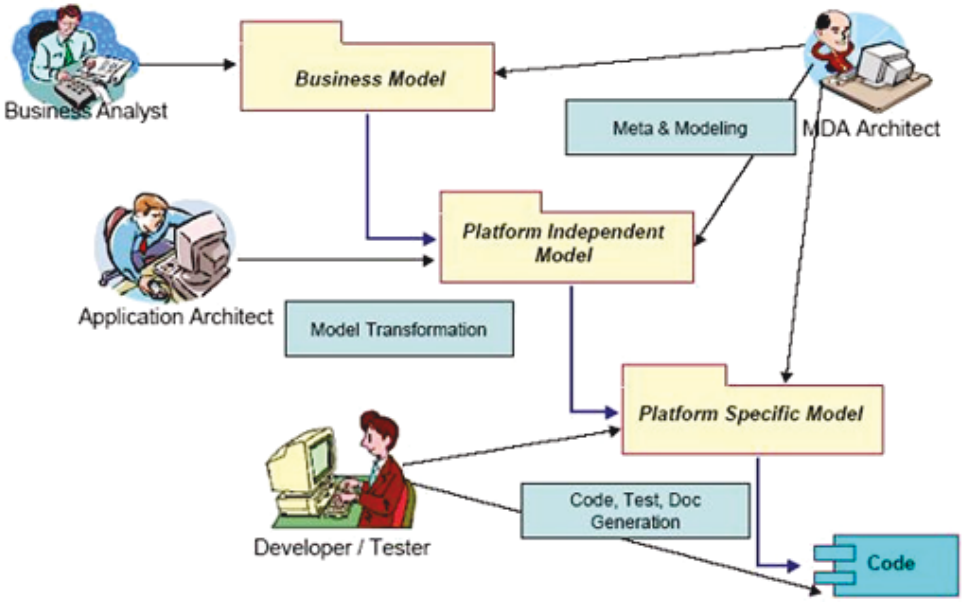

FIGURE 5.1: MDA Development Process.

approach starts with the first model of MDA namely Computational Independent Model (CIM) which describes business ambience and business requirements. From the CIM view, the system is considered as a black box. Figure 5.2 shows the three layer structure of CIM which depicts the artifacts of CIM on the basis of Problem Domain, Application Architecture and Organizational Characteristics. The CIM presents various diagrams that explain various requirements of the enterprise systems, as shown in the figure 5.2, and they are mainly : user, organizational, functional and non functional Requirements. Three UML diagrams depict all these requirements in CIM of MDA approach namely Use Case Diagram, Activity Diagram and Sequence Diagram.

5.2.2 Platform Independent Model

CIM is then transformed to the next model named as Platform Independent Model (PIM) which explicitly explains services and interfaces provided by software system without considering any technology platform. It contains details about business functionality and behavior but no information about the technical details or the platforms on which it may be implemented. A common technique for achieving platform independence is to target a system model for a technology-neutral virtual

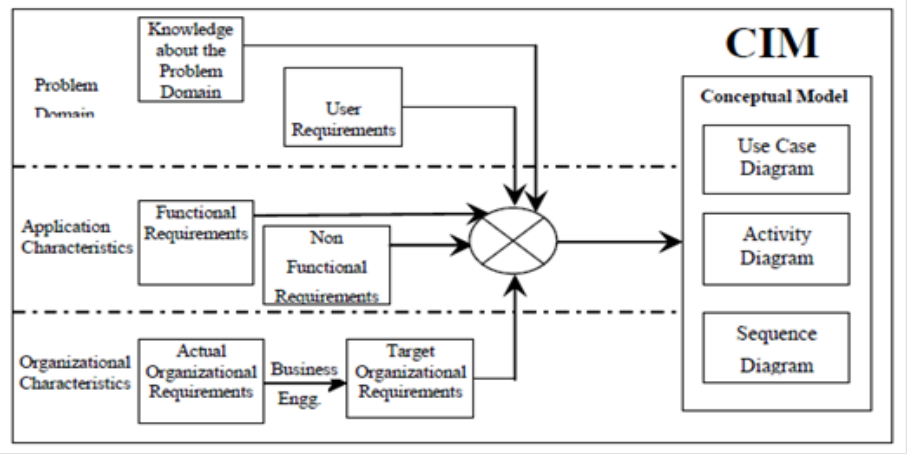

FIGURE 5.2: Computational Independent Model Concepts.

machine [162]. The PIM is further transformed into Platform Specific Model (PSM) for realization of software system to specific technology [161].

5.2.3 Platform Specific Model

PSM is a view of a system from the platform specific viewpoint. A PSM contains the specifications from a PIM but with details about the usage on a concrete platform. PSM usually contains enough information to allow code generation [162]. PDM describes the operation of the system as it uses one or more specific platforms. A PSM might consist of a model from the informational viewpoint, which captures information about the data of a system, and a model from the computational viewpoint, which captures information about the processing of a system, based on a specific platform. As a PSM targets a specific platform, it uses the features of the specific platform specified by a platform model. The PSM corresponds to the specification perspective's design model.

5.2.4 Model Transformation

The figure 5.3 shows the architecture of various models of MDA and transformation among them. Many researchers have proposed various approaches for transforming PIM to PSM in MDA but artifacts of CIM have been somehow ignored by current development in the MDA approach of software development. It is pertinent to

note that to meet the business and user requirements, the importance of CIM and transition from CIM to PIM is critical.

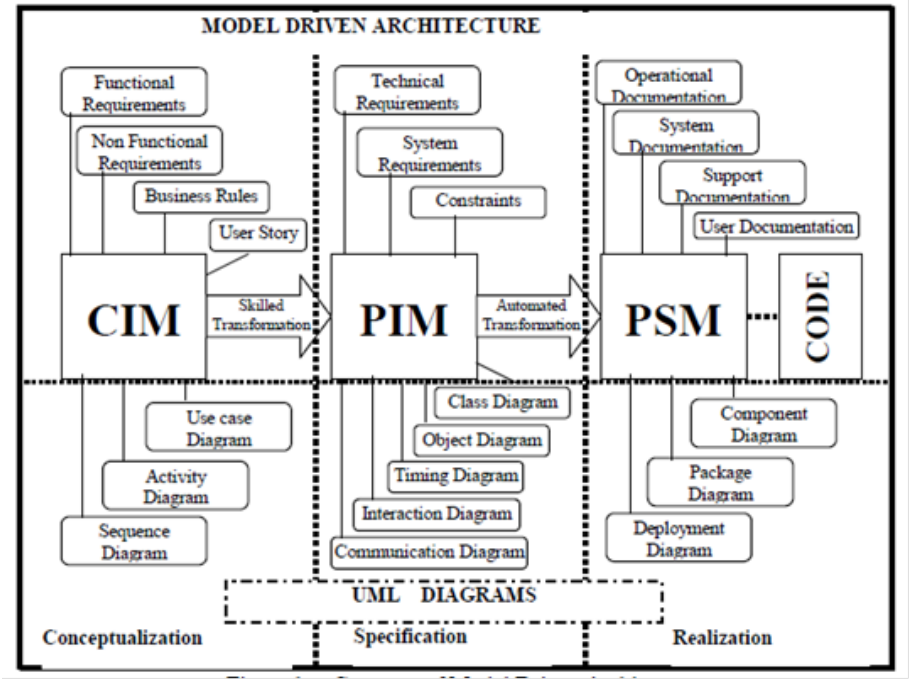

FIGURE 5.3: Concepts of Model Driven Architecture.

5.3 Human-Robot Interaction Scenario

We are interesting to the case of senior citizens who are living on their own. The lack of permanent attention results unfortunately in the late detection of emergency situations. However with the advent of technology, the detection of emergency situations that elderly people may encounter becomes easier. Assistive environments, on one hand, have been established to monitor the senior inhabitant at home for detecting emergency situations. They integrate surveillance devices into the living environments giving remote operators access. Robots become, also, used on the other hand to interact and communicate with humans or other autonomous physical agents by following social behaviours and rules attached to its role. Hence, interaction between humans and robots provide the concept of social robots.

This definition suggests that a social robot must have a physical embodiment (screen characters would be excluded). Recently, some robots have been developed using a screen to display the robot's head. Such a machine is on the borderline of

being a robot. If the body only functions as a holder for the screen then such a system cannot be considered a robot but if the robot has some physical motor and sensor abilities then such a system could be considered as a robot. So, we are in the second category of robots.

Otherwise, privacy of elderly persons raises many ethic and cultural questions. Generally, these people reject the idea of being assisted by a robot. They refuse to be under a video surveillance at home or even to be followed anywhere by a sensor. Unfortunately, without establishing a relationship of trust with the technology, the world cannot change and evolve to the better. Moreover, many home accidents can be avoided if the person was assisted. So, in order to involve this relationship of trust with the technology, we grant to the elderly person the right of setting and defining its own privacy policy rules. Although, it is far from obvious that a privacy management definition could be a simple task.

Nathan is an elderly person suffering from episodic health issues concerning cognitive impairments and blood glucose level. Nathan is in rehabilitation after an ischemic heart attack under the supervision of a health monitoring application that can be connected to a back-end system at the hospital. The monitoring application monitors Nathan's medication and physical exercises for enhancing the oxygenation of his blood and gives access to the physicians in case of emergency or during the scheduled interviews with Nathan.

The monitoring application must have privacy safeguards to avoid the transmission or storage of information concerning the intimate aspects of Nathan's life that can be captured by the robot and the sensors distributed in the Ami environment. The privacy control system should give Nathan the ability to modify a default policy to take into account his privacy requirements that change over time. Basically, the monitoring application has a minimal privacy policy that handles the most important contexts such as in emergency situation. In the latter situation, an alarm must be triggered and a notification should be sent to the emergency department of the hospital. In addition, all the cameras that are in the current location of Nathan must be activated and accessible to the physician, first response people and his family members.

Moreover, during daily living activities, surveillance cameras are disabled by default and the robot can move to any location at home. This robot can follow Nathan

during his walking and can record video using his embedded camera for recognizing Nathan's activities except when Nathan is near to the bathroom, the restroom or when his is sleeping in his bedroom. Consequently, the intrusion alarm is switched on. Nathan can customize this policy in an intuitive manner by giving natural language instructions to the AmI system in natural language. We will present how the proposed meta-model can be valuable for capturing the privacy requirement addressed above and helping in making the design of the monitoring privacy-aware application. We will show, also, the usefulness of the SmartRules rules for setting the privacy policy of Nathan.

5.4 Semantic Privacy Framework at Runtime

5.4.1 Privacy Model for HRI Scenario

In this section, we show how to model privacy policies using the resulting design of our meta-model language. We illustrate it through the the HRI scenario introduced in the previous section. As the first step towards making HRI application for elderly privacy-aware, we extend the abstract concepts of our meta-model. We aim at defining the vocabulary of HRI application by merging both meta-model vocabulary and the application model vocabulary.

Second, we identify the model elements of HRI application representing privacy policies. We determine if such application deals with access control policy, release policy, data handling policy, sensing policy or policy disclosure policy. For this purpose, we must determine which model element we wish to control its access to in the resulting systems. We determine, also, which private resources should be protected, in which context and what are privacy sensitive operations applied on these resources. The figue 5.4 depicts the PIM of our scenario and its class diagram. We distinguish the "Elderly" class that extends the "Owner" class. The Elderly is a member of the "EmergencyFirstAidCommunity". We specify the "Emergency" situation for the PIM and the related private resources such as the health records and emergency notification. In the next step, we define the set of privacy control actions that is mandatory for the every protected resource and define a hierarchy

on actions. Specifically, we fix for each resource type of the dialect the privacy decision as privacy control action or the privacy control process.

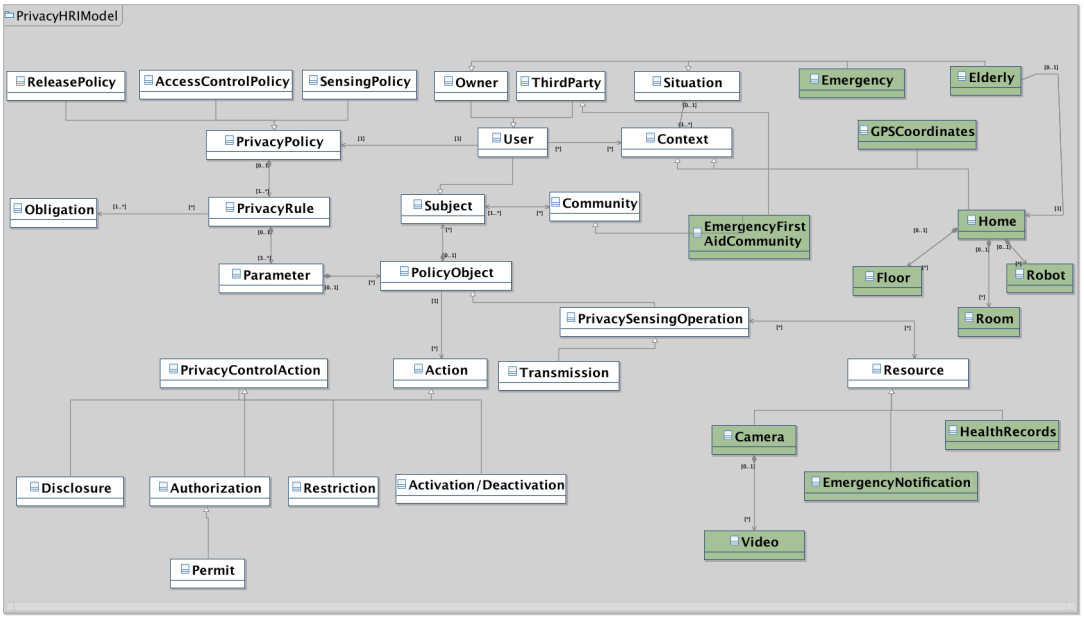

FIGURE 5.4: Human-Robot Interaction Application.

5.4.2 Privacy Policy Model for Daily Living Situations

The aim of this model is to provide the designer with a model that guarantees the elderly privacy protection when interacting with the robot and the ambient intelligence environment in daily living situations. In fact, the policy must control sensitive operations such as the movement of the companion robot, the observations through its sensors and the communication with third parties. These operations have a strong impact on privacy. By default, the privacy control system is setup with a minimal policy, which can be enriched and personalized by the elderly himself or by some one that can act on his behalf. In the case of normal situations, the policy, in its initial setup, allows, for instance, the robot to move to any place at home, follow the elderly and activate the robot camera, except when the latter is busy with some private activities or located in an intimate place such as the bedroom or the bathroom. In the case of an emergency situation, the policy will authorize some operations that were prohibited in normal situation.

In the following sections, we are going to detail the default model of the privacy policy that is applied in normal situations of daily living activities. We will detail

also the privacy policy model that is applied in abnormal situations such as the emergency case. To define this model, we use the privacy policy templates that are introduced in section 4.3.1 of the chapter 4 in association with the concepts of the meta-model. For the sake of space and clarity, we provide also the privacy rules in SmartRules language.

For the daily living situations scenario, we consider the following communities : family community, friends community and emergency first aid community. This latter is considered as a subconcept of the trusted third party. It is a composition of persons that are members of the family community or agents working for medical organization and having one of the following roles : First_Aid_People, Emergency_Nurse, Emergency_Physician or Family_Physician.

5.4.2.1 Privacy Policy in the Normal Situations of Daily Living

Social communication is a daily living activity that has a strong impact on privacy when the elderly is in the situation of communicating remotely with friends, relatives and his physician using chatting service of the robot. In this situation, the policy allows the system enabling the robot's camera as well as the transmission and the disclosure of the elderly context parameters. The video scene captured by the robot as a part of the context parameters is disclosed. Before the transmission of these parameters, the system must obfuscate the faces of other persons that may be present in the video scenes. With respect to the social communication with physician, the policy can be customized to allow the system to transmit the vital signals recorded by medical devices. Vital signals are also a part of the context parameters of the elderly. The physician's remote system that receives the context parameters information must execute a data-anonymization action when storing these information in the database. Such an action is considered as an obligation of the privacy policy.

To design this policy, we need to define, with the SmartRules language, the following privacy sensitive operations : the transmission of situation's context parameters, the remote observation with robot sensors, and the enabling or the disabling of the robot or of its sensors. Hence, we consider for this policy the following privacy rules :

Privacy Rule 1. In the context of daily living activities, enabling the robot to run as shown in the figure 5.5.

Rule 1 : DefaultEnablingRobot

Conditions

?situation := Situation (?my_Context := hasContext,
?my_activity:= isInstanceOf (ActivityOfDailyLiving),

exists (?situation) ;
?robot :=Robot() ;
?sensitive_operation := isInstanceOf (Modification());
?private_resource := ?robot;
?sensitive_operation.object:= ?private_resource;

Actions

?privacy_control_action := Authorization();
?privacy_control_action→object:= ?sensitive_operation;

update (?privacy_control_action)

Enable ?_Enable := createAction(Enable)
?_Enable→object := ?private_resource;
execute(?_ Enable);
End

FIGURE 5.5: Enabling the Robot during Daily Living Activities.

Privacy Rule 2. During chatting with friends and relatives, the policy authorizes the robot enabling the camera and following the elderly when he is moving. Then, it discloses his context and the video scene captured by the robot. In this context, the policy allows the robot to follow the elderly anywhere at home except when inhabitant is near to or inside the bathroom or the restroom. The corresponding privacy rule is depicted in the figure 5.6.

Privacy Rule 3. Prohibition of the observation with the robot camera and disable the robot when it is near the person current location and the latter is an intimate place such as the bathroom or the restroom except when emergency. This privacy rule is depicted in the figure 5.7.

Rule 2 : SocialCommunicationActivity

Conditions

?situation := Situation (?my_Context := hasContext,
?my_activity:= isInstanceOf (SocialCommunicationActivity));
exists (?situation) ;

?robot :=Robot() ;
?sensitive_operation := isInstanceOf (Observation());
?private_resource := Robot_Camera ();
?sensitive_operation.object:= ?private_resource;

Actions

?privacy_control_action := Authorization();
?privacy_control_action→object:= ?sensitive_operation;

update (?privacy_control_action)

Enable ?_Enable := createAction(Enable)
?_ Enable→object := ?private_resource;
execute(?_ Enable);

End

FIGURE 5.6: Enabling the Robot Camera during Social Communication Activities.

Privacy Rule 4. Allowing members of elderly's family community to know his situation when he is at healthcare center or at his physician office. This privacy rule is depicted in the figure 5.8.

5.4.2.2 Privacy Policy in Emergency Situation

In the case of emergency situation, an alarm must be triggered and the notification should be sent to the emergency department of the hospital. Hence, all the cameras are activated at the current location. In this context, we need to define several privacy policies to control two privacy sensitive operations, and they are (i) the transmission of the emergency notification and (ii) the remote observation of the scene where emergency has occurred.

Rule 3 : ProhibitionObservation

Conditions

?situation := Situation (?my_Context := hasContext,
?my_Location:= isInstanceOf (IntimateSpaceRegion));
?robot :=Robot(?robot_location:=isNearToPerson);

exists (?situation, ?robot);
not exists (?situation := isInstanceOf (Emergency())

?sensitive_operation := isInstanceOf (Observation());
?private_resource := Robot_Camera ();
?sensitive_operation.object:= ?private_resource;

Actions

?privacy_control_action := Prohibition();
?privacy_control_action→object:= ?sensitive_operation;
update (?privacy_control_action)
Disable ?_Disable := createAction(Disable)
?_Disable→object := ?private_resource;
execute(?_Disable) ;
StopNavigation ?_StopNavigation:= createAction(StopNavigation);
?_StopNavigation→object := ?robot;
execute(?_StopNavigation);

End

FIGURE 5.7: Disabling the Robot Camera and Prohibition of the Observations in Intimate Area.

The privacy control actions are the following : (i) permitting or denying the transmission and the observation of the private resource and (ii) disclosing the emergency notification content and its context to individuals of the emergency first aid community.

The private resource is the emergency notification. The situation is the emergency, which is composed of the following contextual parameters : location of the monitored subject "elderly" such as the indoor coordinates, the room, the floor, the GPS coordinates and also the visual scene captured by the available video cameras, the

Rule 4 : SituationTransmission

Conditions

?situation := Situation (?my_Context := hasContext,
?my_Location:= isInstanceOf (HealthcareSpaceRegion));
exists(?situation);
?private_resource := ?situation;
?sensitive_operation := isInstanceOf (Transmission);
?sensitive_operation→object:= ?private_resource;
?authorized_third_party:= Familly();
?sensitive_operation.destination := one (?authorized_third_party);

Actions

?privacy_control_action := authorization();
?privacy_control_action.object:= ?sensitive_operation;
update (?privacy_control_action);

End

FIGURE 5.8: Discolure of the Elderly Outdoor Situation to the Family Members Community.

robot location, the luminosity and patient health records. Hence, we consider the following rules :

Privacy Rule 1. The aim of this rule is making control when allowing or denying the transmission of the emergency notification to each members of the Emergency_First_Aid_Community. This rule can be written informally as : If an emergency situation occurs then authorize the transmission of the Private Resource (emergency notification) to trusted and authorized people. To define this rule with the SmartRules language, we use the authorization policy template and the concepts of the proposed meta-model as depicted in the figure 5.9.

Privacy Rule 2. To define this rule, we use a release policy template and the concepts of the proposed meta-model. A Release policy that specifies what are the private resources to be disclosed and to whom. In this case, the emergency notification content and inhabitant location and his/her current health status. The third party is the first aid people and patient relatives. The corresponding privacy rule is depicted in the figure 5.11.

Rule: EmergencyNotificationTransmission

Conditions

?situation := Emergency ();
?private_resource := Emergency_Notification ();
?sensitive_operation := Transmission();
?authorized_third_party:= Emergency_First_Aid_Community();

Actions

?sensitive_operation.destination := one (?authorized_third_party);
?privacy_control_action := authorization();
?privacy_control_action.object := ?sensitive_operation;
?sensitive_operation.object := ?private_resource;
update (?privacy_control_action)
End

FIGURE 5.9: Authorization of Transmission of the Emergency Notification in the Emergency Situation.

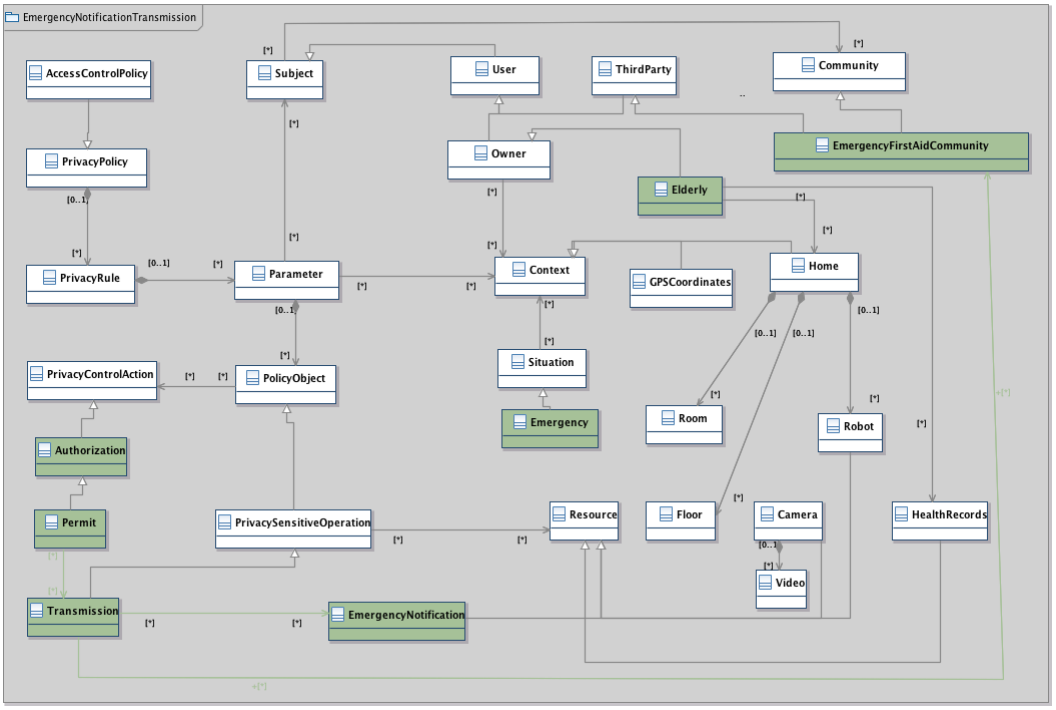

FIGURE 5.10: UML diagram for the Emergency Notification Transmission.

Privacy Rule 3. Activating the robot camera to capture the scene of the emergency situation. The aim of this rule is to activate the camera and allow the

Rule 2 : EmergencyNotificationDisclosure

Conditions

?situation := Emergency ();
?sensitive_operation := Transmission();
?privacy_control_action := Authorization();
?privacy_control_action→object := ?sensitive_operation;
?private_resource := Emergency_Notification ();
?monitored_person := Context(?location:=locatedAt(),
?video:= capturedVideoSceneURI);

Actions

?privacy_control_action := Disclose();
?privacy_control_action→object:= ?private_resource;
?privacy_control_action→object:= ?location;
?privacy_control_action→object:= ?video;
update (?privacy_control_action);

End

FIGURE 5.11: Controlling the Disclosure of the Emergency Notification.

streaming of the video scene. By default, the robot camera is disabled when the robot is located in highly private places such as the bedroom and the bathroom. When the emergency situation occurs, the system enables the robot camera and authorizes the observation operation in the private region of the indoor space. To define this rule, we use a release policy template and the concepts of the proposed meta-model as follows. We note that, in this case, we focus on the access to the camera by the first aid people it is done through the disclosure of the video uri in rule 2. The corresponding privacy rule is depicted in the figure 5.13.

5.5 Description of the LISSI's Ubiquitous Platform

The experiment of the policy management rules is undertaken by using the ubistruct living lab infrastructure of the LISSI research laboratory, which is composed of

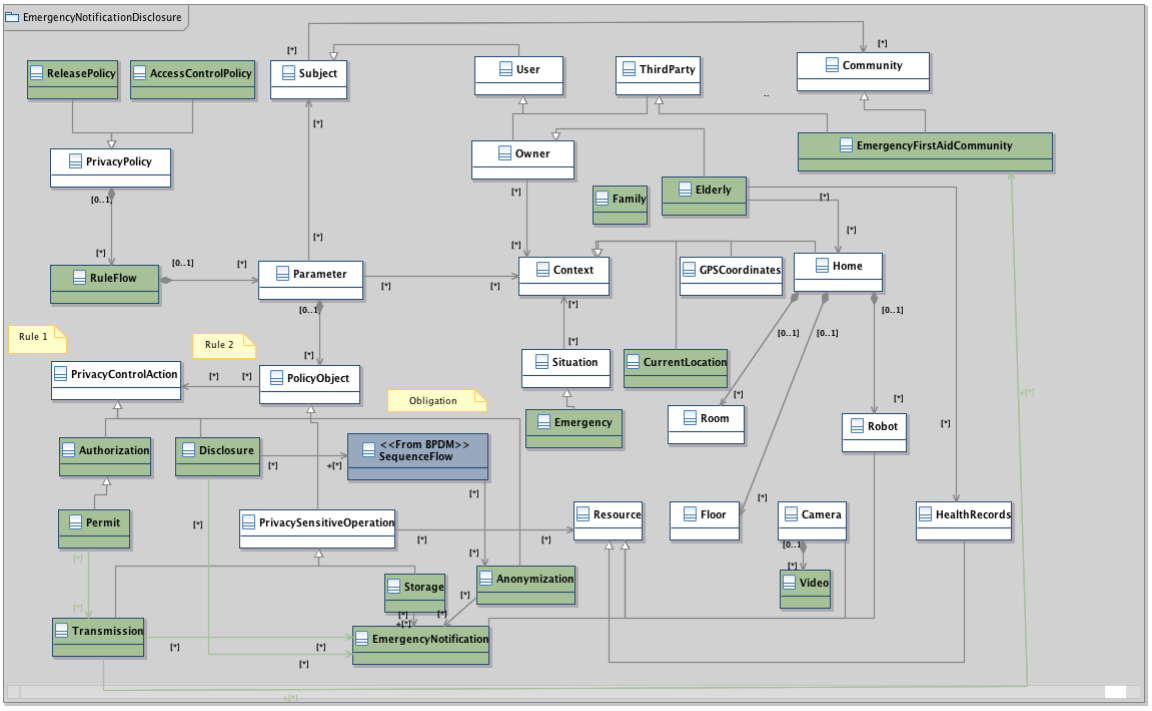

FIGURE 5.12: UML diagram for the Disclosure of Emergency Notification.

Rule 3 : ActivateCamera

Conditions

?situation := Emergency();
?sensitive_operation := Observation();
?private_resource := Robot_Camera ();
?privacy_control_action→object:= ?sensitive_operation;
?monitored_person_context := exists (Context(
?location:= one(isPrivateRegion)));

Actions

?privacy_control_action := enable();
?privacy_control_action→object:= ?private_resource;
update (?privacy_control_action);

End

FIGURE 5.13: Authorization of the First Aid People to Access to the Robot Camera In Emegrency Situation.

hardware and servers that host the applications and web services. The architecture of living lab is modular and can be reconfigured to meet the different requirements of experimentation and scenarios, thanks to the use of a variety of wireless, mobile furniture and equipments that can be found in the market. The latter range from wireless sensor networks and actuators to smart devices such as smartphones, tablets and the mobile robot Kompai from Robotsoft. The hardware infrastructure is described in the following and depicted in the figure 5.14.

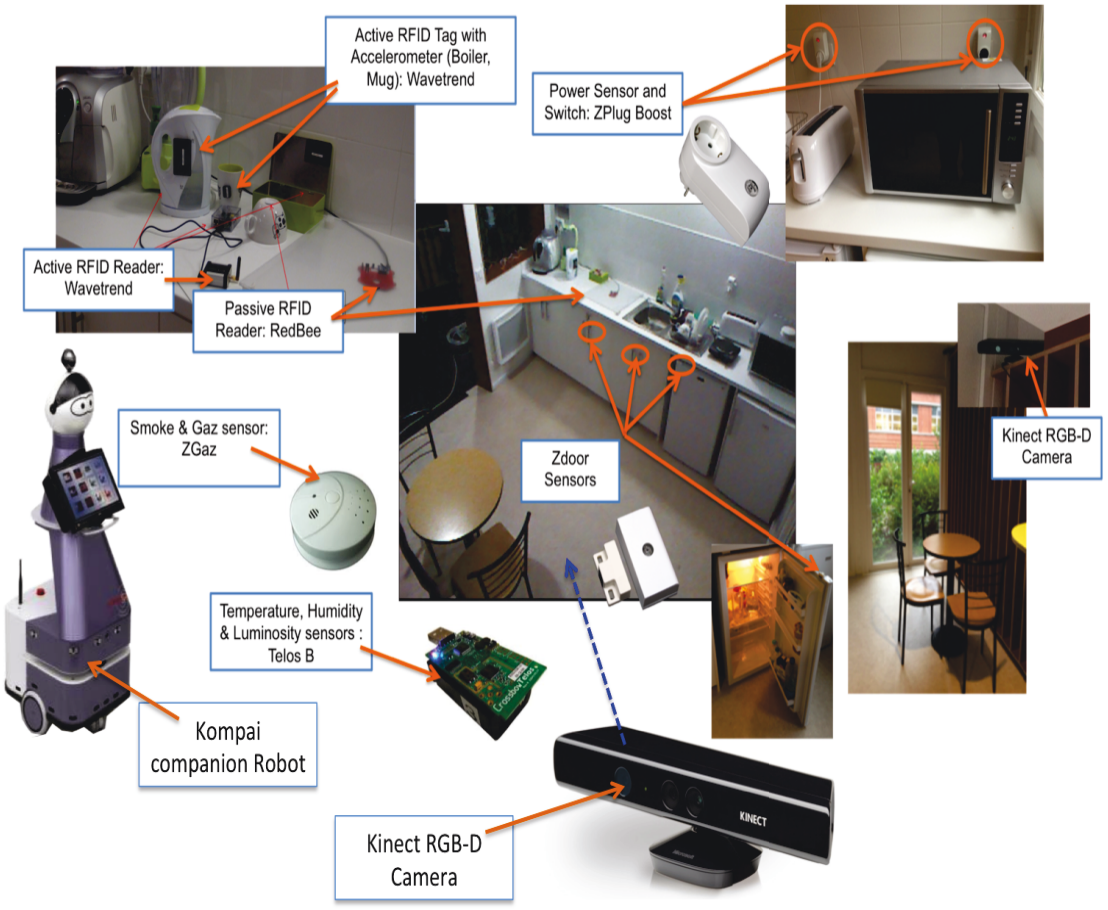

FIGURE 5.14: The Ubistruct Living Lab Infrastructure of the LISSI's Platform.

- **Kompai robot :** It was developed by the Robosoft company. It has several sensors and actuators to ensure the essential functions of navigation and interactions with the environment and support users in their daily tasks as shown in the figure 5.14. Kompai is equipped with two cameras, a tablet PC, a 2D laser radar, ultrasonic sensors for obstacle detections, infra-red sensors, contact sensors, two motors for controlling the wilds. The robot platform is remotely monitored through the urbi middleware and web services.

The Kompai robot embeds, also, several functionalities ranging from a low-level control to high-level assistive services of end users that are based on medication calendar, social communication with Skype, e-mail management and voice recognition.

- **Power management :** The ZPlug Boost is a wireless Power Outlet which can be switched On/Off by Zigbee™ protocol. It can commute 16A devices on 220V/230V. It measures the instantaneous power and cumulated consumption of the connected device.

- **Presence detection and identification :**

  1. Active RFID : The RX202 from Wavetrend provides instant reporting of all detected Wavetrend active RFID tags. It allows user configurable tag data and read range filters. It is installed on the Kompai Mobile Robot and on Rasberry Pi modules through USB connection. The latter can be deployed in any location in the living space.
  2. Passive RFID : The RoboticsConnection RedBeeTM RFID Reader is a sophisticated reader that can work in standalone, or Networked BPAN (Broadcast Personal Area Network) mode. The reader is designed to work with all EM41xx family 125 kHz RFID tags including cards, buttons, capsules, disks, key fobs, and others. A wireless connexion can established between the The RedBeeTM reader and Robot or Mobile Raspberry Pi module by using XBee Zigbee wireless module, which acts as a wireless serial interface.
  3. Cricket : It is indoor location system for that is used by any agent to track humans or objects position accurately. A socket server provides fine-grained location information of Cricket beacons : space region identifiers, position coordinates X,Y and Z, and orientation. The Cricket native location computing system have been improved to correct deployment and measurement drawbacks. The Cricket beacons are installed on the mobile robot and Raspberry Pi to track respectively the robot and human positions in the ambient space.

- **Environment sensing :**

1. Doors opening and closing detection : The ZDoor is very low power Zigbee™ wireless sensor. It detects opening and closing of doors or windows with a magnet and reed-switch mechanism. ZDoor is compliant with Zigbee Pro 2007 stack and can be easily add in an existing network. This sensor is installed on cupboards doors, fridge as well as doors and windows of the living environment.
2. Measuring ambient temperature, humidity and luminosity : The TelosB mote platform is an open source, low-power wireless sensor module that allows measuring RSSI, temperature, temperature, humidity and luminosity. The TelosB is compliant with IEEE 802.15.4. TelosB runs a Contiki embedded operating system.
3. Motion detection : ZMove is a Zigbee™ passive infrared sensor (PIR sensor) that measures infrared (IR) light radiating from objects in its field of view in the ambient space. Motion detection events are send as alarms to the central node by the Zigbee™ wireless network.

5.6 Conclusion

In this chapter, we have implemented our semantic framework according to the MDA process development for the Human-Robot interaction Scenario. Through this particular application domain of ambient assisted living and social interactions through ubiquitous robots, we have shown the usefulness for the privacy by design approach that allows better defining and handling privacy policies and control procedures from the design stage of AmI applications.

In addition, the privacy rules expressed in the SmartRules language in this part of the thesis are expressed from the ontology defined for the domain application of HRI the scenario. These rules use, hence, the ontology vocabulary described with the language µConcept. They handle knowledge of privacy configurations such as the user context/situation that are continually changing. The SmartRules language enables expressing simple privacy rules that have the ability to change the values of concepts instances checking all the constraints defined in the semantic model. It aims, also, at applying actions on concepts instances.

The notion of variables in the language of SmartRules rules is similar to those used in other rule languages, such as Drools, Jess, etc. This variable can be used to store knowledge (instance, property value or literal). However, unlike languages requiring the declaration of the type of the variable, the SmartRules language automatically infers the type of the variable of the referenced item. Declared in a rule, variables are visible only within this rule. However, it is not possible to declare a variable with the same name already declared as a global variable.

Chapter 6

Conclusion

The objective of this thesis is to address a main problem which has so far not received much attention in privacy research : the privacy by design problem. We aim at allowing any ubiquitous system designer implementing easily mechanisms to manage privacy policies. In order to address this research problem, we first studied prominent privacy solutions for ambient intelligence that were suggested in the last decades. First, we introduced the main concepts of our research domain which is Ambient Intelligence. In particular, we shed light on the Ubiquitous/Pervasive Computing environments. Then, we highlighted the main applications in the AmI environments. We focused on applications for the elderly as they present a high-risk population regarding their privacy. Hence, we defined several privacy challenges that we took into consideration when using AmI services to better protect users' personal data. These challenges can be summed up in the semantic policies, the multi-domain interoperability, fine grained privacy control, the management of obligations, context awareness, adaptability and the conflicts management. Afterwards, we studied the existing solutions and approaches for managing user privacy in the AmI applications. Then, we summarized the analysis of these different approaches based on the challenges criteria listed above.

The results of this analysis are important for us to determine the way to design our privacy system in AmI environment. Hence, our thesis brings an extension of the traditional system-centric approach of managing privacy in Ambient Intelligence towards a more user-centric approach that is based on policies that can be defined, customized and handled by the users themselves. The main contribution of the

thesis is the proposition of a semantic framework for privacy by design that allows for better defining and handling privacy policies and control procedures from the design stage of AmI applications. To this end, we adopted the MDE approach and proposed a meta-model for specifying and implementing privacy policies. The conceptual constructs of the meta-model are also based on ontology language constructs to bring more expressiveness and allow for the formal description and reasoning on privacy policies according to the closed world assumption. This combination is important, on the one hand, for enabling the full interoperability of privacy controls and policies between the different applications and, on the other hand, it offloads application designers from implementing privacy policy management operations.

In addition, this framework allows for defining privacy policies according to a meta-model and generic templates allows for an interoperable and coherent control for the protection of user's privacy over heterogeneous third party domains. The templates are used for implementing policies as inference rules in the SmartRules language. The latter are defined with the concepts of the meta-model and run on forward chaining inference engines according the closed world assumption. Unlike some approaches that are based on the SWRL inference language, the Smart Rules languages allow for using variables on concepts, which are mandatory for defining adaptive policies and rules with negation. These features are not supported in SWRL. The proposed meta-model allows for defining privacy policies that can be tailored to individuals privacy needs and brings a fine-grained access control over the private resources. The proposed meta-model simplifies the definition of privacy policies by proposing a restricted set of upper concepts, which have roots in the Ontology Definition Meta-model (ODM) and Business Process Definition Meta-Model (BPDM). Privacy policies can be defined according to specific templates that allow for controlling the release and handling of private resources and the policies themselves as well as the observations of sensors in private regions. OCL constraints are defined to bring model-theoretic semantics of the classes definition in the meta-model to avoid the definition of wrong privacy policies.

The second contribution of this thesis is the proposition of a generic middleware for implementing the privacy management models as inference rules by using the concepts of the meta-model according to the proposed policy template. These rules can be handled by forward-chaining inference engines. A proof of concept dealing

with privacy in ambient assisted living and social interactions through ubiquitous robots have been implemented to validate the proposed approach.

We do believe that this thesis is only an extensive snapshot of our work in progress. We plan to continue with some of the threads of future work that we can identify in short, medium and long term.

In the short term, we are planning to complete the implementation of Human-Robot Interaction scenario. We will focus mainly on the related privacy middleware for this application. We also plan to specify more privacy rules in the SmartRules language. Furthermore, we plan to test our framework with further case studies and scenarios. Most importantly, we are interested in privacy case studies for multiple domains, and also at different stages of systems development according to the Model-Driven Architecture (MDA) development process. Then, for our proposed framework, we plan to evaluate each analysis criteria that we have detailed in the third chapter. That's why, we expect to collaborate with stakeholders for experimenting privacy in such applications.

In the medium term, we are planning to validate and verify the framework with the Alloy analyser and mainly the privacy meta-model to detect their flaws. Thanks to Alloy, we will validate if our definitions of privacy policy are consistent and compatible with each other. So, we plan to create an Alloy model for privacy meta-model. We also plan to define the architecture of our Alloy privacy model, the rules of this model, the translation of the privacy concepts definitions into Alloy and the results of the verification.

Finally, in the long term, we aim at translating the privacy rules into the natural language. This will enable users to easily define and modify their privacy policies. Our plan is to use the Natural Rule Language (NRL) to provide a user-friendly alternative to languages like OCL, XSLT, XPath and many others, particularly in scenarios where they would be considered too technical.

Bibliography

[1] Mark Weiser. The computer for the Twenty-First Century. Scientific American, 265(3):94–104, 1991.

[2] Sachin Singh, Sushil Puradkar, and Yugyung Lee. Ubiquitous computing: connecting pervasive computing through semantic web. Inf. Syst. E-Business Management, 4(4):421–439, 2006. URL http://dblp.uni-trier.de/db/journals/isem/isem4.html#SinghPL06.

[3] Debashis Saha and Amitava Mukherjee. Pervasive computing: A paradigm for the 21st century. Computer, 36(3):25–31, March 2003. ISSN 0018-9162. doi: 10.1109/MC.2003.1185214. URL http://dx.doi.org/10.1109/MC.2003.1185214.

[4] Kalle Lyytinen and Youngjin Yoo. Issues and Challenges in Ubiquitous Computing - Introduction. Commun. ACM, 45(12):62–65, December 2002. ISSN 0001-0782. doi: 10.1145/585597.585616. URL http://dx.doi.org/10.1145/585597.585616.

[5] Vaskar Raychoudhury, Jiannong Cao, Mohan Kumar, and Daqiang Zhang. Middleware for pervasive computing: A survey. Pervasive and Mobile Computing, September 2012. ISSN 15741192. doi: 10.1016/j.pmcj.2012.08.006. URL http://dx.doi.org/10.1016/j.pmcj.2012.08.006.

[6] Diane J. Cook and Sajal K. Das. Review: Pervasive computing at scale: Transforming the state of the art. Pervasive Mob. Comput., 8(1):22–35, February 2012. ISSN 1574-1192. doi: 10.1016/j.pmcj.2011.10.004. URL http://dx.doi.org/10.1016/j.pmcj.2011.10.004.

[7] Roy Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, and M. Dennis Mickunas. Towards security and privacy for pervasive computing. In Proceedings of the 2002 Mext-NSF-JSPS international conference on Software security: theories and systems, pages 1–15, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-00708-3.

[8] Margaret Morris, Jay Lundell, Eric Dishman, and Brad Needham. New perspectives on ubiquitous computing from ethnographic study of elders with cognitive decline. pages 227–242. 2003. URL http://www.springerlink.com/content/2dmxcejn5c4hdhra.

[9] P. J. Brown, J. D. Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. Personal Communications, IEEE [see also IEEE Wireless Communications], 4(5):58–64, 1997.

[10] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A Survey on Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4):263–277, 2007.

[11] Jason Pascoe. Adding generic contextual capabilities to wearable computers. pages 92–99, 1998.

[12] Anind K. Dey. Context-aware computing: The cyberdesk project. In AAAI 1998 Spring Symposium on Intelligent Environments, pages 51–54, Palo Alto, 1998. AAAI Press. URL http://www.cc.gatech.edu/fce/cyberdesk/pubs/AAAI98/AAAI98.html.

[13] Albert van Breemen, Xue Yan, and Bernt Meerbeek. icat: an animated user-interface robot with personality. In Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, AAMAS '05, pages 143–144, New York, NY, USA, 2005. ACM. ISBN 1-59593-093-0. doi: 10.1145/1082473.1082823. URL http://doi.acm.org/10.1145/1082473.1082823.

[14] David Feil-seifer and Maja J Matari. Ethical principles for socially assistive robotics, 2010.

[15] I. Berlin. Two concepts of liberty. In his Four Essays on Liberty, Oxford University Press, pages 118–172, 1979.

[16] Philip Brey. Freedom and privacy in ambient intelligence. Ethics and Infomration. Technology., 7(3):157–166, September 2005. ISSN 1388-1957. doi: 10.1007/s10676-006-0005-3. URL http://dx.doi.org/10.1007/s10676-006-0005-3.

[17] David Feil-seifer and Maja J Matari. Defining socially assistive robotics. In in Proc. IEEE International Conference on Rehabilitation Robotics (ICORR05, pages 465–468, 2005.

[18] H. Kozima and C. Nakagawa. Social robots for children: practice in communication-care. In Advanced Motion Control, 2006. 9th IEEE International Workshop on, pages 768–773, 2006. doi: 10.1109/AMC.2006.1631756.

[19] Cory D. Kidd. A sociable robot to encourage social interaction among the elderly. In in: International Conference on Robotics and Automation, 2006.

[20] A. Chibani, Y. Amirat, S. Mohammed, E. Matson, N. Hagita, and M. Barreto. Ubiquitous robotics: Recent challenges and future trends. Robotics and Autonomous Systems, 2013. URL http://www.sciencedirect.com/science/article/pii/S0921889013000572.

[21] Olaf Hartig, Martin Kost, and Johann-Christoph Freytag. Automatic Component Selection with Semantic Technologies. In Proceedings of the 4th International Workshop on Semantic Web Enabled Software Engineering (SWESE) at ISWC, October 2008.

[22] Douglas C. Schmidt. Model-driven engineering. IEEE Computer, 39(2), February 2006. URL http://www.truststc.org/pubs/30.html.

[23] Carol C. Burt, Barrett R. Bryant, Rajeev R. Raje, Andrew M. Olson, and Mikhail Auguston. Model driven security: Unification of authorization models for fine-grain access control. In 7th International Enterprise Distributed Object Computing Conference (EDOC 2003), 16-19 September 2003, Brisbane, Australia, Proceedings, pages 159–173. IEEE Computer Society, 2003. ISBN 0-7695-1994-6. doi: http://csdl.computer.org/comp/proceedings/edoc/2003/1994/00/19940159abs.htm.

[24] Michael Hecker, Tharam S. Dillon, and Elizabeth Chang. Privacy ontology support for e-commerce. IEEE Internet Computing, 12(2):54–61, 2008. ISSN 1089-7801. doi: http://doi.ieeecomputersociety.org/10.1109/MIC.2008.41.

[25] Markus Schumacher. 6. toward a security core ontology. In Security Engineering with Patterns, volume 2754 of Lecture Notes in Computer Science, pages 87–96. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40731-7. doi: 10.1007/978-3-540-45180-8_6. URL http://dx.doi.org/10.1007/978-3-540-45180-8_6.

[26] L. Fratiglioni, H.X. Wang, K. Ericsson, M. Maytan, and B. Winblad. Influence of social network on occurrence of dementia: a community-based longitudinal study. Lancet, 355(9212):1315–9, 2000.

[27] Margaret Morris, Jay Lundell, and Proactive Health. Ubiquitous computing for cognitive decline: Findings from intel's proactive health research, intel corporation, 2003.

[28] Danah Boyd and Nicole B. Ellison. Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication, 13(1-2), November 2007.

[29] Kimberly Banquil. Social Networking Sites Affect One's Academic Performance Adversely. PhD thesis, 2009.

[30] Sonia Ben Mokhtar, Liam McNamara, and Licia Capra. A middleware service for pervasive social networking. In Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, M-PAC '09, pages 2:1–2:6, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-849-0.

[31] Sonia B. Mokhtar and Licia Capra. From pervasive to social computing: Algorithms and deployments. In Proceedings of the ACM International Conference on Pervasive Services (ICPS 2009), July 2009.

[32] CharuC. Aggarwal, Naveen Ashish, and Amit Sheth. The internet of things: A survey from the data-centric perspective. In Charu C. Aggarwal, editor, Managing and Mining Sensor Data, pages 383–428. Springer US, 2013. ISBN 978-1-4614-6308-5. doi: 10.1007/978-1-4614-6309-2_12. URL http://dx.doi.org/10.1007/978-1-4614-6309-2_12.

[33] J H Fowler and N A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. Bmj Clinical Research Ed., 337:a2338, 2008.

[34] Social robots laboratory, institut für informatik, germany, 2013. URL http://srl.informatik.uni-freiburg.de/home.

[35] Fariba Sadri. Ambient intelligence: A survey. ACM Comput. Surv., 43(4): 36:1–36:66, October 2011. ISSN 0360-0300. doi: 10.1145/1978802.1978815. URL http://doi.acm.org/10.1145/1978802.1978815.

[36] Samuel D. Warren and Louis D. Brandeis. The right to privacy. Harward Law Review, 4(5):193–220, December 1890.

[37] Penders and Jacques. Privacy in (mobile) telecommunications services. Ethics and Inf. Technol., 6(4):247–260, 2004.

[38] Siani Pearson. Taking account of privacy when designing cloud computing services. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09, pages 44–52, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-1-4244-3713-9. doi: 10.1109/CLOUD.2009.5071532. URL http://dx.doi.org/10.1109/CLOUD.2009.5071532.

[39] Louis Harris '& Associates and Alan F. Westin. Commerce, communication and privacy online. New York: Louis Harris&Associates, 1997. URL http://p3pbook.com/links.html.

[40] Sen. John Edwards. Location privacy protection act of 2001. 2001. URL http://www.techlawjournal.com/cong107/privacy/location/s1164is.asp.

[41] Victoria Bellotti and Abigail Sellen. Design for privacy in ubiquitous computing environments. In ECSCW'93: Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work, pages 77–92. Kluwer Academic Publishers, 1993.

[42] Johann Cas. Privacy in pervasive computing environments - a contradiction in terms? In Technology and Society Magazine, IEEE, pages 24–33, 2005.

[43] Ryan Babbitt, Johnny Wong, and Carl Chang. Towards the modeling of personal privacy in ubiquitous computing environments. In COMPSAC '07: Proceedings of the 31st Annual International Computer Software and Applications Conference, pages 695–699, Washington, DC, USA, 2007. IEEE Computer Society.

[44] Scott Lederer, Jason I. Hong, Anind K. Dey, and James A. Landay. Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing, 8(6):440–454, November 2004.

[45] Giovanni Iachello and Jason Hong. End-user privacy in human-computer interaction. Found. Trends Hum.-Comput. Interact., 1(1):1–137, 2007. ISSN 1551-3955.

[46] Marc Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. pages 273–291. Springer.

[47] Michael Friedewald, Elena Vildjiounaite, Yves Punie, and David Wright. Privacy, identity and security in ambient intelligence: A scenario analysis. Telematics and Informatics, 24(1):15–29, February 2007. doi: 10.1016/j.tele.2005.12.005. URL http://linkinghub.elsevier.com/retrieve/pii/S0736585305000778.

[48] Marc Langheinrich. Personal Privacy in Ubiquitous Computing – Tools and System Support. PhD thesis, ETH Zurich, Zurich, Switzerland, may 2005.

[49] Dan Hong, Mingxuan Yuan, and Vincent Y. Shen. Dynamic privacy management: a plug-in service for the middleware in pervasive computing. In Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, MobileHCI '05, pages 1–8, New York, NY, USA, 2005. ACM.

[50] Eric S. Chung, Jason I. Hong, James Lin, Madhu K. Prabaker, James A. Landay, and Alan L. Liu. Development and evaluation of emerging design patterns for ubiquitous computing. In Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques, DIS '04, pages 233–242, New York, NY, USA, 2004. ACM. ISBN 1-58113-787-7.

[51] The brave new world of ambient intelligence: A state-of-the-art review, 2006. URL http://is.jrc.ec.europa.eu/pages/TFS/documents/SWAMI_D1_Final_001.pdf.

[52] Alison George. Living online: The end of privacy? New Scientist, 2569, 2006. URL http://www.newscientist.com/channel/tech/mg19125691.700-living-online-the-end-ofprivacy.

[53] Janet Kornblum and Mary B. Marklein. What you say online could haunt you. USA Today, Tuesday 2006. URL http://www.usatoday.com/tech/news/internetprivacy/2006-03-08-facebook-myspace_x.htm.

[54] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks (The Facebook case). In Proceedings of the 2005 ACM workshop on Privacy in the electronic society, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM.

[55] Alessandro Acquisti and Ralph Gross. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In Privacy Enhancing Technologies, volume 4258 of Lecture Notes in Computer Science, chapter 3, pages 36–58. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006.

[56] Catherine Dwyer, Starr R. Hiltz, and Katia Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In Proceedings of the Thirteenth Americas Conference on Information Systems, August 2007. URL http://aisel.aisnet.org/amcis2007/339.

[57] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. Commun. ACM, 50:94–100, October 2007. ISSN 0001-0782.

[58] Sören Preibusch, Bettina Hoser, Seda Gürses, and Bettina Berendt. Ubiquitous social networks - opportunities and challenges for privacy-aware user modelling. In Proceedings of Workshop on Data Mining for User Modeling, 2007. URL http://vasarely.wiwi.hu-berlin.de/DM.UM07/Proceedings/05-Preibusch.pdf.

[59] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. Mobile cloud computing: A survey. Future Gener. Comput. Syst., 29(1):84–106, January

2013. ISSN 0167-739X. doi: 10.1016/j.future.2012.05.023. URL http://dx.doi.org/10.1016/j.future.2012.05.023.

[60] M. Friedewald, O. Raabe, P. Georgieff, D.J. Koch, and P. Neuhäusler. Ubiquitäres Computing:. Studien des Büros für Technikfolgen-Abschätzung beim Deutschen Bundestag. Edition Sigma, 2010. ISBN 9783836081313. URL http://books.google.sn/books?id=UxeWAWrPr1sC.

[61] Yudistira Asnar, Paolo Giorgini, and John Mylopoulos. Goal-driven risk assessment in requirements engineering. Requir. Eng., 16(2):101–116, June 2011. ISSN 0947-3602. doi: 10.1007/s00766-010-0112-x. URL http://dx.doi.org/10.1007/s00766-010-0112-x.

[62] Privacy impact assessment (pia). URL http://www.ico.org.uk/for_organisations/data_protection/topic_guides/privacy_impact_assessment.

[63] Ann Cavoukian. Privacy by Design: The 7 Foundational Principles, Mai 2010. Revised: Oktober 2010.

[64] Qingfeng He, Annie I. Anton, For Healthcare [hip, Gramm Leach, and Bliley Act. A framework for modeling privacy requirements in role engineering, 2003.

[65] Tom Gruber. Ontology. In Encyclopedia of Database Systems, pages 1963–1965. 2009.

[66] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. Computer, 29(2):38–47, February 1996. ISSN 0018-9162. doi: 10.1109/2.485845. URL http://dx.doi.org/10.1109/2.485845.

[67] Ravi Sandhu, Ravi S, and Qamar Munawer. How to do discretionary access control using roles, 1998.

[68] Nora Cuppens-Boulahia, Frédéric Cuppens, Diala Abi Haidar, and Hervé Debar. Negotiation of prohibition: An approach based on policy rewriting. In Sushil Jajodia, Pierangela Samarati, and Stelvio Cimato, editors, SEC, volume 278, pages 173–187. Springer, 2008. URL http://dblp.uni-trier.de/db/conf/sec/sec2008.html#Cuppens-BoulahiaCHD08.

[69] Carol Geyer. Xacml access control markup language ratified as oasis open standard. 2003. URL http://www.oasis-open.org/news/pr/xacml-access-control-markup-language-ratified-as-oasis-open-standard.

[70] Torsten Lodderstedt, David A. Basin, and Jürgen Doser. Secureuml: A uml-based modeling language for model-driven security. In Proceedings of the 5th International Conference on The Unified Modeling Language, UML '02, pages 426–441, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44254-5. URL http://dl.acm.org/citation.cfm?id=647246.719477.

[71] OMG. Object constraint language, May 2006.

[72] Siv Hilde Houmb and Kine Kvernstad Hansen. Towards a uml profile for security assessment. In UML 2003, Workshop on Critical Systems Development with UML, page 815829, 2003.

[73] Jan Jürjens. Towards development of secure systems using umlsec, 2001.

[74] John Krumm. A survey of computational location privacy. Personal Ubiquitous Comput., 13(6):391–399, August 2009. URL http://dx.doi.org/10.1007/s00779-008-0212-5.

[75] Model-Driven Architecture: Vision, Standards And Emerging Technologies.pdf (application/pdf Object). 2001. URL http://www.cwmforum.org/Model-DrivenArchitecture.pdf.

[76] OMG. Object management group, inc. URL http://www.omg.org/.

[77] Omg model-driven architecture home page:, 2001. URL http://www.omg.org/mda/index.htm.

[78] Omg. Model driven architecture - a technical perspective, July 2001. URL http://www.omg.org/docs/ormsc/01-07-01.pdf.

[79] D. D'Souza. Model-Driven Architecture and Integration - Opportunities and Challenges, Version 1.1, Kineticum. 2001. URL http://www.catalysis.org/publications/papers/2001-mda-reqs-desmond-6.pdf.

[80] OMG. Mda guide version 1.0.1, 2003. URL http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf.

[81] OMG. Catalog of omg specifications, 2003. URL http://www.omg.org/technology/documents/index.htm.

[82] OMG. Unified modeling language, 2010. URL http://www.omg.org/spec/UML/2.4/.

[83] OMG. Omg's metaobject facility, 2011. URL http://www.omg.org/spec/MOF/.

[84] OMG. Xml metadata interchange (xmi), 2010. URL http://www.omg.org/spec/XMI.

[85] OMG. Uml profile for enterprise distributed object computing specification, 2005. URL http://doc.omg.org/ptc/02-02-05.

[86] L. Hart, P. Emery, B. Colomb, K. Raymond, S. Taraporewalla, D. Chang, Y. Ye, E. Kendall, and M. Dutra. OWL Full and UML 2.0 Compared. Technical report, 2004.

[87] OMG. Ontology definition metamodel (odm), 2009. URL http://www.omg.org/spec/ODM/1.0/.

[88] Jean Bézivin. Model Driven Engineering: An Emerging Technical Space. In Ralf Lämmel, João Saraiva, and Joost Visser, editors, Generative and Transformational Techniques in Software Engineering, volume 4143 of Lecture Notes in Computer Science, chapter 2, pages 36–64. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-45778-7. doi: 10.1007/11877028_2. URL http://dx.doi.org/10.1007/11877028_2.

[89] Douglas C. Schmidt. Model-driven engineering. IEEE Computer, 39(2), February 2006. URL http://www.truststc.org/pubs/30.html.

[90] Jay Bhalodi Stan Kurkovsky, Oscar Rivera. Classification of privacy management techniques in pervasive computing. International Journal of u- and e- Service, Science and Technology, 1:55–72, 2007. URL http://www.sersc.org/journals/IJUNESST/vol1no1/papers/07.pdf.

[91] W3C. Platform for privacy preferences (p3p) project. 2002. URL http://www.w3.org/P3P/.

[92] Lorrie Faith Cranor and Lawrence Lessig. Web Privacy with P3p. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2002.

[93] Lorrie Faith Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The platform for privacy preferences 1.0 (p3p1.0). 2002. URL http://www.w3.org/TR/P3P.

[94] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. An xpath-based preference language for p3p. In Proceedings of the 12th international conference on World Wide Web, WWW '03, pages 629–639, New York, NY, USA, 2003. ACM.

[95] Rc policy workbench project, 2005. URL http://sourceforge.net/projects/jrc-policy-api.

[96] The netscape archive, 2000. URL http://browser.netscape.com.

[97] AT&T Corporation. Find web sites that respect your privacy, 2006. URL http://www.privacybird.org.

[98] P3pbuilder, 2004. URL http://www.p3pbuilder.com.

[99] P3p policy tools, 2004. URL http://www.webentrust.com/p3p.html.

[100] M. Zuiderweg. A p3pbased privacy architecture for a contextaware services platform. Master's thesis, University of Twente, Enschede, the Netherlands, 2003.

[101] Gnter Karjoth and Matthias Schunter. A privacy policy model for enterprises. 2002.

[102] Michael Backes, Birgit Pfitzmann, and Matthias Schunter. A toolkit for managing enterprise privacy policies. In In Proc. of ESORICS03, LNCS 2808, pages 162–180. Springer, 2003.

[103] Matthias Schunter, Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. Enterprise privacy authorization language (epal 1.1). 2003. URL http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/,2003.

[104] Michael Backes, Walid Bagga, Gnter Karjoth, and Matthias Schunter. Efficient comparison of enterprise privacy policies. In In SAC 04: Proceedings of the 2004 ACM symposium on Applied computing, pages 375–382. ACM Press, 2004.

[105] Enterprise privacy authorization language (epal 1.2), 2003. URL http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/.

[106] Adam Barth, John C Mitchell, and Justin Rosenstein. Conflict and combination in privacy policy languages. Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, page 45, 2004. URL http://portal.acm.org/citation.cfm?id=1029195.

[107] Ryan Wishart, Karen Henricksen, and Jadwiga Indulska. Context privacy and obfuscation supported by dynamic context source discovery and processing in a context management system. In Jadwiga Indulska, Jianhua Ma, Laurence Tianruo Yang, Theo Ungerer, and Jiannong Cao, editors, UIC, volume 4611 of Lecture Notes in Computer Science, pages 929–940. Springer, 2007. URL http://dblp.uni-trier.de/db/conf/uic/uic2007.html#WishartHI07.

[108] Prime - privacy and identity management for europe. URL https://www.prime-project.eu.

[109] Claudio Agostino Ardagna, Sabrina De Capitani di Vimercati, and Pierangela Samarati. Enhancing user privacy through data handling policies. In Ernesto Damiani and Peng Liu, editors, DBSec, volume 4127 of Lecture Notes in Computer Science, pages 224–236. Springer, 2006. URL http://dblp.uni-trier.de/db/conf/dbsec/dbsec2006.html#ArdagnaVS06.

[110] Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001. ISBN 0471389226.

[111] Morris Sloman. Policy driven management for distributed systems. Journal of Network and Systems Management, 2:333–360, 1994.

[112] The web application security consortium. 2000. URL http://www.cgisecurity.com.

[113] Rbac. URL http://csrc.nist.gov/groups/SNS/rbac/.

[114] Roy. Towards security and privacy for pervasive computing. In In Proceedings of International Symposium on Software Security, pages 1–15, 2002.

[115] Axiomatics. URL http://www.axiomatics.com.

[116] Barbara Carminati, Elena Ferrari, and Andrea Perego. Rule-based access control for social networks, volume 4278, pages 1734–1744. Springer, 2006.

[117] Mohammad Mannan and Paul C. van Oorschot. Privacy-enhanced sharing of personal content on the web. In Proceeding of the 17th international conference on World Wide Web, WWW '08, pages 487–496, New York, NY, USA, 2008. ACM.

[118] Dr. Carrie and E. Gates. Access control requirements for web 2.0 security and privacy. In Proc. of Workshop on Web 2.0 Security & Privacy (W2SP 2007, 2007.

[119] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: a temporal role-based access control model. In Proceedings of the fifth ACM workshop on Role-based access control, RBAC '00, pages 21–30, New York, NY, USA, 2000. ACM. ISBN 1-58113-259-X. doi: http://doi.acm.org/10.1145/344287.344298. URL http://doi.acm.org/10.1145/344287.344298.

[120] James B. D. Joshi, Elisa Bertino, Usman Latif, and Arif Ghafoor. Trbac: A temporal role -based access control model. ACM Transactions on Information and System Security (TISSEC, 2001.

[121] Maria Luisa Damiani, Elisa Bertino, Barbara Catania, and Paolo Perlasca. Geo-rbac: A spatially aware rbac. ACM Trans. Inf. Syst. Secur., 10, February 2007. ISSN 1094-9224.

[122] Ogc standards and specifications, 1999. URL http://www.opengeospatial.org/standards.

[123] Or-bac. URL http://orbac.org/.

[124] Hervé Debar, Yohann Thomas, Frédéric Cuppens, and Nora Cuppens-Boulahia. Enabling automated threat response through the use of a dynamic security policy. Journal in Computer Virology, 3(3):195–210, 2007.

[125] Qun Ni. Privacy-aware Role-Based Access Control. PhD thesis, 5 2010.

[126] Qun Ni, Alberto Trombetta, Elisa Bertino, and Jorge Lobo. Privacy-aware role based access control. In Proceedings of the 12th ACM symposium on Access control models and technologies, SACMAT '07, pages 41–50, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-745-2. doi: http://doi.acm.org/10.1145/1266840.1266848. URL http://doi.acm.org/10.1145/1266840.1266848.

[127] Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombeta. Privacy-aware role-based access control. ACM Trans. Inf. Syst. Secur., 13:24:1–24:31, July 2010. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/1805974.1805980. URL http://doi.acm.org/10.1145/1805974.1805980.

[128] The enterprise privacy authorization language(epal 1.1). IBM Zurich Research Laboratory,Switzerland., 2003. URL http://www.zurich.ibm.com/security/enterprise-privacy/epal.

[129] Robert Johnson, Michael Hart, and Amanda Stent. More content – less control: Access control in the Web 2.0. In Proceedings of the Workshop on Web 2.0 Security and Privacy at the IEEE Symposium on Security and Privacy, 2007. URL http://www.amandastent.com/papers/JohnsonHartStent2007.pdf.

[130] Susana Alcalde Bag´ues, Luis A. Ramon Surutusa, Mikel Arias, Carlos Fernandez-Valdivielso, and Ignacio R. Matias. Personal privacy management for common users. 3:89–106, 2009.

[131] K. Sheikh, M. Wegdam, and M.J. van Sinderen. Quality-of-context and its use for protecting privacy in context aware systems. Journal of Software, 3 (3):83–93, March 2008. URL http://doc.utwente.nl/62252/.

[132] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, J. Polk, and J. Rosenberg. Common Policy: A Document Format for Expressing Privacy Preferences, February 2007. URL http://www.ietf.org/rfc/rfc4745.txt.

[133] Jason Cornwell, Ian Fette, Gary Hsieh, Madhu Prabaker, Jinghai Rao, Karen Tang, Kami Vaniea, Lujo Bauer, Lorrie Cranor, Jason Hong, Bruce Mclaren,

Mike Reiter, and Norman Sadeh. User-controllable security and privacy for pervasive computing. In In Eighth IEEE Workshop on Mobile Computing Systems and Applications (HotMobile, 2007.

[134] Jaehong Park and Ravi Sandhu. The uconabc usage control model. ACM Trans. Inf. Syst. Secur., 7(1):128–174, February 2004. ISSN 1094-9224. doi: 10.1145/984334.984339. URL http://doi.acm.org/10.1145/984334.984339.

[135] Slim Trabelsi, Laurent Gomez, and Yves Roudier. Context-aware security policy for the service discovery. In AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, pages 477–482, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2847-3.

[136] Rei : A policy specification language. URL http://rei.umbc.edu.

[137] Jeffrey Bradshaw Andrzej Uszok. Kaos tutorial. URL http://ontology.ihmc.us/KAoS/KAoS%20Tutorial.pdf.

[138] A framework for semantic web policies. URL http://ceur-ws.org/Vol-401/iswc2008pd_submission_13.pdf.

[139] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham. Rowlbac: representing role based access control in owl. In Proceedings of the 13th ACM symposium on Access control models and technologies, SACMAT '08, pages 73–82, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-129-3. doi: 10.1145/1377836.1377849. URL http://doi.acm.org/10.1145/1377836.1377849.

[140] Murat Sensoy, Timothy J. Norman, Wamberto Weber Vasconcelos, and Katia P. Sycara. Owl-polar: Semantic policies for agent reasoning. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang 0007, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, International Semantic Web Conference (1), volume 6496 of Lecture Notes in Computer Science, pages 679–695. Springer, 2010. ISBN 978-3-642-17745-3. URL http://dblp.uni-trier.de/db/conf/semweb/iswc2010-1.html#SensoyNVS10.

[141] Hassan Takabi and James B. D. Joshi. Semantic-based policy management for cloud computing environments. IJCC, 1(2/3):119–144, 2012. URL http://dblp.uni-trier.de/db/journals/ijcc/ijcc1.html#TakabiJ12.

[142] Lalana Kagal. Rei : A Policy Language for the Me-Centric Project. Technical report, HP Labs, September 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-270.html.

[143] Sparql query language for rdf. URL http://www.w3.org/TR/rdf-sparql-query/.

[144] Owl reasoning examples. URL http://owl.man.ac.uk/2003/why/latest.

[145] Pellet: Owl 2 reasoner for java. URL http://clarkparsia.com/pellet.

[146] Dibyajyoti Ghosh, Anupam Joshi, Tim Finin, and Pramod Jagtap. Privacy control in smart phones using semantically rich reasoning and context modeling. In IEEE Symposium on Security and Privacy Workshops, pages 82–85. IEEE Computer Society, 2012. ISBN 978-1-4673-2157-0. URL http://dblp.uni-trier.de/db/conf/sp/spw2012.html#GhoshJFJ12.

[147] Madan Oberoi, Pramod Jagtap, Anupam Joshi, Tim Finin, and Lalana Kagal. Information integration and analysis: A semantic approach to privacy. In SocialCom/PASSAT, pages 959–965, 2011.

[148] Pramod Jagtap, Anupam Joshi, Tim Finin, and Rosa Laura Zavala Gutierrez. Preserving privacy in context-aware systems. In ICSC, pages 149–153. IEEE, 2011. ISBN 978-1-4577-1648-5. URL http://dblp.uni-trier.de/db/conf/semco/icsc2011.html#JagtapJFG11.

[149] Palanivel Andiappan Kodeswaran and Evelyne Viegas. Towards a privacy preserving policy based infrastructure for social data access to enable scientific research. In Eighth Annual Conference on Privacy, Security and Trust, pages 103–109. IEEE, August 2010.

[150] Palanivel Andiappan Kodeswaran and Evelyne Viegas. Applying Differential Privacy to Search Queries in a Policy Based Interactive Framework. In ACM International Workshop on Privacy and Annonymity for Very Large Datasets. ACM, November 2009.

[151] Prajit Kumar Das, Tim Finin, and Anupam Joshi. Energy efficient sensing for managing context and privacy on smartphones. In Proceedings of the Workshop on Society, Privacy and the Semantic Web – Policy and Technology, October 2013. held in conjunction with the 2013 International Semantic Web Conference.

[152] Ryan Babbitt, Johnny Wong, and Carl Chang. Towards the modeling of personal privacy in ubiquitous computing environments. In Proceedings of the 31st Annual International Computer Software and Applications Conference - Volume 02, COMPSAC '07, pages 695–699, Washington, DC, USA, 2007. IEEE Computer Society.

[153] GAUTHAM V. PALLAPA. A Privacy Enhanced Situation-aware Middleware Framework For Ubiquitous Computing Environments. PhD thesis, 12 2009. URL http://hdl.handle.net/10106/4867.

[154] Dan Brickley and R.V. Guha. Rdf vocabulary description language 1.0: Rdf schema, 2004. URL http://www.w3.org/TR/rdf-schema/.

[155] Guus Schreiber Mike Dean. Owl web ontology language reference, 2004. URL http://www.w3.org/TR/owl-ref/.

[156] Graham Moore Lars Marius Garshol. Topic maps - data model, 2008. URL http://www.isotopicmaps.org/sam/sam-model/.

[157] OMG. Ontology definition metamodel (odm), 2009. URL http://www.omg.org/spec/ODM/1.0/.

[158] Carlos Damásio, Anastasia Analyti, Grigoris Antoniou, and Gerd Wagner. Supporting Open and Closed World Reasoning on the Web. pages 149–163. 2006. doi: 10.1007/11853107_11. URL http://dx.doi.org/10.1007/11853107_11.

[159] Lyazid Sabri, Abdelghani Chibani, Yacine Amirat, and Gian piero Zarri. Semantic reasoning framework to supervise and manage contexts and objects in pervasive computing environments. 2013 27th International Conference on Advanced Information Networking and Applications Workshops, 0:47–52, 2011. doi: http://doi.ieeecomputersociety.org/10.1109/WAINA.2011.148.

[160] Gian Piero Zarri. Nkrl, a knowledge representation tool for encoding the ‘meaning’ of complex narrative texts. Nat. Lang. Eng., 3(2): 231–253, September 1997. ISSN 1351-3249. doi: 10.1017/S1351324997001794. URL http://dx.doi.org/10.1017/S1351324997001794.

[161] Yashwant Singh and Dr. Manu Sood. Article: The impact of the computational independent model for enterprise information system development. International Journal of Computer Applications, 11(8):21–26, December 2010. Published By Foundation of Computer Science.

[162] Igor Sacevski and Jadranka Veseli. Introduction to model driven architecture (mda). Seminar Paper, June 2007.