Undefined 0 (0) 1
IOS Press

Special Issue: Ambient Agent

Requirement Analysis Abstractions for AmI System Design

Patrizia Ribino [a,*], Massimo Cossentino [a], Carmelo Lodato [a], Salvatore Lopes [a], and Valeria Seidita [a,b]
[a] *Istituto di Calcolo e Reti ad alte Prestazioni - Consiglio Nazionale delle Ricerche - Italy*
[b] *Dip. di Ingegneria Chimica, Gestionale, Informatica, Meccanica - Universitá degli Studi di Palermo - Italy*

Abstract. Current trends in the AI's evolution are going towards enriching environments with intelligence in order to support humans in their everyday life. AmI systems are plunged in the real world and humans expect to interact with them in a way that is similar to the one they have with other humans. In this kind of systems, where eliciting requirements involves several documents and stakeholders (mainly users that will be the first consumers of the system), the requirement analysis phase can be affected by incomplete, ambiguous and imprecise information. Hence, the need to find a fruitful way for knowledge management and its representation at design time. In this paper we propose a set of abstractions to be used during the early requirements analysis of AmI systems development. The result is a simple and at the same time powerful set of concepts and guidelines for providing environment knowledge representation for AmI systems.

Keywords: Requirement Analysis, Smart Environment, Ontology, AmI Model, Software Design

1. Introduction

The attention paid to system users, today more than ever, leads to the need of creating systems able to interact with users in an intelligent way but, above all, to systems able to work considering the user as the centre of the system itself. The new idea that is emerging in recent years is that the intelligent interaction between users and system does not involve a single device, that is a computer or some kind of programmable control unit. User does not more only interact with the system through software and then devices such as keyboard and so on. The system, with its hardware and software part, is now distributed/integrated in the environment the user lives in. In this new meaning, the environment becomes intelligent and it is able to perceive user-related data and process them (reason above, learn, etc.) in order to support the user in all aspects of its life within a given environment. The research area and applications related to the presented scenario are known with the term Ambient Intelligence (AmI). "*A digital environment that proactively but sensibly supports people in their daily lives*" - this is how J.Augusto defines AmI systems in [4]. We may sum up that main features of AmI systems are: the capability of being aware of the context, of adapting and evolving and finally the capability of learning. These are still open research issues. Engineering such complex systems requires novel design and development approaches including phases where peculiar aspects of AmI systems are modeled. The interaction between system and users must be designed to meet needs and requirements of users. This paper addresses how to support the development of AmI systems more specifically focusing on identifying the right abstractions to be managed during the requirement analysis and the design activities to be performed. Thus, we propose a portion of a design process relating to the Early Requirements Analysis phase for capturing the requirements of AmI systems. To this

*Corresponding Author: P.Ribino Email:ribino@pa.icar.cnr.it

end, we firstly have to define the model of a generic intelligent environment.

The proposed AmI model grounds on two key elements: Smart and Dumb entities. The former are intentional and rational entities endowed with some reasoning mechanism, they populate the environment and make it intelligent. The latter are environment resources without any kind of "intelligence". We know that the concept of intelligence is very complex to represent, hence using few elements in order to capture it is surely restrictive. In this work, we do not claim to cover every aspect concerning the intelligence of a system. We talk about the intelligence of the system in terms of a triple of features owned by smart entities: intentionality, rationality and knowledge. Moreover, our model is also based on organizational principles and normative aspects that allow to manage complex intelligent environments.

In this perspective, the proposed Early Requirement analysis phase aims at coping with all the activities necessary for defining elements and features of the AmI system. In particular, a portion of the design process we propose is based on an ontological representation of the AmI system from the problem viewpoint. We believe that ontological representations may help in identifying some key elements useful for the requirement analysis and, as a consequence, for the design of the solution of a generic system. Indeed in [24] we presented a goal oriented requirements analysis based on the ontology description of the problem domain. We establish our thoughts on the results of several studies that assert ontologies may have a significant role in the model driven engineering and may offer several benefits to a design process (first of all to the requirement analysis)[3][27][7][26]. Only to name a few: ontologies are useful for formalizing knowledge and for disambiguation of terms; they help in better understanding requirements and in eliminating redundancies and ambiguities; they simplify comprehension among stakeholders and is useful for making clear the stakeholders knowledge; they lets the designer create categories for the elements in the domain problem thus allowing to identify the artifacts that will compose the environment. From these considerations, we think that an ontological-based design process could bring significant benefits and could be more effective in the field of AmI systems where the environment plays a central role. The environment in fact is a huge source of "knowledge" that requires some kind of representation in order to be managed. This knowledge concerns not only the elements the environment is composed of but also the intricate relationships among these elements and the constraints that limit their relationships.

The question this paper wants to answer is: what are the abstractions to use for developing AmI systems?.

The main contribution introduced in this paper is the creation of two tasks along with the related guidelines and techniques for performing the Early Requirements analysis of AmI systems. The adoption of these new tasks provide some advantages such as:

- A deep understanding of the physical environment that in a AmI system is fundamental.
- The disambiguation and improvement of the problem domain comprehension and consequently the improvement of requirements elicitation due to the use of an ontological formalization.
- The identification of some fundamental abstractions for AmI systems analysis: (i) Smart and Dumb entities; (ii) Intentional and Unintentional Actions; (iii) Rules.
- The possibility to apply guidelines for goal identification based on ontological patterns we presented in [24].
- Guidelines for Roles/Capacities identification based on ontological representations of the problem domain.

In order to prove these claims we will show a case study of AmI system located in an industrial context. This case study revealed the necessity to manage some entities that have been well represented with the abstractions we have introduced in our metamodel. Moreover, the use of the problem ontology as exchange artifact among stakeholders has allowed the adoption of a common language and improved the problem specification. As well as the attention paid to the collection of technical specifications of the various components of the system has allowed us to avoid design errors that would have a great impact on implementation costs.

The remainder of the paper is organized as follows. Section 2 provides an overview on the state of the art. Section 3 defines the fundamental abstractions we identified for designing AmI systems. Section 4 and 5 respectively detail the proposed Early Requirements Analysis phase and a proposal for completing the Requirements Analysis phase. Finally, sections 6 and 7 illustrate the case study, our discussions and conclusions.

2. State of the Art and Related Works

In the definition given by J.Augusto in [4] and introduced in Section 1, with the term *sensibly* he means the capability to learn to get familiarity and to get in empathy with the user in the same way a nurse, a trainer or a butler does. To act with sensibility, both in the case of humans and of digital devices, requires intelligence in order to provide the right support. In early years, the basic idea of Ambient Intelligence research was how to add devices and technologies to existing environment in order to realize the proactive behavior and the sensibility said before [8].

In the latest years the growing diffusion of mobile and fixed connectivity and the diffusion of mobile devices has determined a new computational scenario in which users may access resources and/or services or interact each others in every moment or in every place. In this scenario a new trend of research in the area of AmI is oriented towards the development of applications conceived for pervasive and ubiquitous environments able to provide results according to the context such us for example the presence and position of users, their preferences, available resources, objects populating the environment and interacting more or less autonomously with the users, etc.

Several definitions of Ambient Intelligence are reported in [25][9][14][1][4]; one we found covering all the main concepts related to AmI is:

> "ambient intelligence concerns a software system able to be sensitive, adaptive and responsive to changes in the physical environment. One of the main aim of Ambient Intelligence is to build systems that make environment be context aware in order to fulfill users' requirements" [14]

From these definitions it descends that AmI includes and is related to many areas in computer science: sensors, network, interfaces, ubiquitous and pervasive computing and artificial intelligence. However none of them covers the whole AmI. In fact, AmI prescribes a new way of seeing technology for supporting users. In the latest years great advancements have been made in the state of the art in all that concerns technological aspects of AmI but still a lots has to be made in the areas of software engineering, conceptualizations and standards for AmI. Nevertheless, "possessing the necessary supporting technology is not enough"[9]. For instance, multi-agent paradigm and multi-agent systems are very useful for modeling real-world and social systems but actually they provide mechanisms and abstractions for modeling and simulating entities such as home, room, cars, etc., hence entities that are instances of an environment [23].

We think that the Requirement Analysis phase of AmI applications needs ways for modeling environment and that we have to consider all the entities involved in the "intelligent" environment and not only the environment itself as a whole; in our knowledge software engineering, and AOSE, doe not provide such means by now. A very important point of view on the principles of design in ubiquitous and pervasive computing, which can be reported in the AmI area, is the one proposed by Brooks in [6]; he identifies the who, where, when, what and why aspects to be taken into account during the design phase of ubiquitous and pervasive systems, in a few words: *who*: the user of the system, the role he plays and the relationships with other users and all the other elements living in the environment; *where*: the position of each user or object of the environment; *when*: the system's dynamic, hence all the user (or other elements) activities that influence the environment and its states change; *what*: activities and tasks have to be identified and recognized in order to provide the right support to users; *why*: understanding intentions and objectives in response to users activities in order to anticipate and meet their needs.

On the base of these principles J. Augusto in [4] restates the lack of techniques, standards and so on for designing AmI systems and identifies the main components of such systems. They compose a triplet <E, IC, I>, Environment, Interaction Constraints and Interactors; the Environment is composed of all the elements living in the environment and is described by means of an ontology, the Interaction Constraints describes all the possible way the Environment and the Interactors may interact and Interactors is the set of "beneficiaries" that interact but is outside the system.

Hence, J. Augusto underlines the necessity of representing the AmI system by means of the environment that he describes through an ontology. There are three different approaches in literature for modeling context: ontology, attribute and value and CC/PP (W3C) extension. We think that representing environment through ontology is one of the most useful way for representing the richness of environment in AmI. However, we found Augusto's approach too general for our purposes and for implementing the Brook's quintet.

In our work, we also use ontology for representing the environment and we go beyond the triplet Augusto proposes. We propose a metamodel containing the elements of the problem domain, with regards to an AmI application that have to be modeled and repre-

sented during the activities of the Requirements Analysis phase. As it can be seen in the following section, we considered a detailed set of elements for representing the users of the systems, for all objects involved in the environment state changes and the related action, either intentional or unintentional.

Another approach to AmI system design is the one proposed by A. Coronato et al. in [11], here the authors propose a formal method for representing requirements of AmI systems, especially for high risk systems, and for reducing errors during the Requirements Analysis phase. They also had the need of representing the domain entity through an ontology and a glossary in order to remove ambiguity. Our approach goes a little beyond in the domain representation by extending the ontology description to the whole environment which, it is our conviction, has to be modeled at the beginning of the requirement analysis phase.

A well known model of environment has been proposed by Ferber in [16]. Such model describes an enviroment mainly made of Objects, Agents and Operations. Objects are situated and passive entities. Agents are active entities that by means of Operations are able to act on objects. In this paper we propose a more extended representation of an environment than the one proposed by Ferber although they share some similarities. In fact we also consider an environment made of Passive Entities (resembling Ferber's objects) and Artificial Beings or Systems (such as cognitive agents) considered as a category of Smart Entities able to perform Actions. In addition, our model provides further abstractions in order to consider new entities and also to differentiate entities according to their capacities, in the Ferber model they are represented in the same way. Only to name a few, we introduce the Users that are key elements of an AmI System, they purposely interact with other elements and have to be modeled during the analysis of an AmI system. Moreover, we have differentiated the active entities in Dumb and Smart Entities. Dumb Entities are active entities operating in response to a request or event without performing any kind of reasoning (printer, air conditioner, reactive agent, etc...). Smart entities instead are cognitive and intentional entities owning knowledge about the world on which they can reason and they have objectives motivating their actions.

As regards design methodologies, several methodologies provide means for modeling the environment in general and some of them may be useful for modeling intelligent environments. We found in literature two main contributions, in both of them we found evidences of the need for modeling the entities involved in the environment.

The authors of INGENIAS [21] made significant advancements in their methodology to meet AmI concepts and extended the metamodel in order to address AmI systems development. It is worth noting that the specificity of the application context could not be faced using the INGENIAS classical metamodel abstractions so in [17] they added some new important elements like for instance the *context* for dealing with the way humans behave in the domain. These new elements are used for creating a visual modeling language focused on interactions among context and agents.

Another important work comes from Weyns, in [28] he integrates the environment into his model for situated agents as a design abstraction overcoming, in so doing, the problem of simply considering environment as infrastructure. He models environment through a three levels representation thus providing support to agents that can use the environment for achieving their goals. Nevertheless, we think that the rationale behind his work is considering the environment as a set of resources, we instead aim to model the environment as a whole intelligent entity able to interact with humans; agents that are identified during the design phase are plunged in the environment.

In [20] the authors propose a methodology for developing multi-agent systems for transportation domain, which is a context where modeling environment is of fundamental importance. In their approach the agent roles sense the environment through services but there is not an explicit model of the environment.

To sum up, due to their features, AmI systems, more than others, have to be faced both under the technological and the methodological aspects; the two are highly related and influence each others. The former is out of the scope of our work but we consider it for identifying the right way for modeling all the involved entities in the environment as we will explain in the next section.

3. The Proposed Abstractions for AmI System Requirements Analysis Phase

In this section we present the Problem Domain Metamodel we adopt for performing the Requirements Analysis phase for AmI Systems.

In order to deduce generalized abstractions for modeling the environment of AmI systems, we observed that they are often inserted in an already existing environment made of persons and objects that influence,

and are influenced by, the new AmI system. Moreover, AmI systems commonly own a strong physical part that needs to be modeled, particularly in the case the system has to be deployed in a preexisting environment. Hence, appropriate abstractions to be used in specific analysis models may help to address possible issues that may emerge both during integration of several heterogeneous components and from constraints that the physical configuration of preexisting environment may impose. The development of an intelligent environment may be highly dependent on its physical part. Our point of view is sketched in Fig.1, the solution environment is the AmI system that has to be designed for a specific need. For instance, a house enriched with an AmI system for providing services like shutting off the light or the television when a person leaves a room. The room presents a predefined/existing layout, what we call *Existing Environment*, in this case: walls, windows, electrical plugs, television and air-conditioning. All these objects have to be enriched with new environment elements, such has sensors and actuators, and with the software part of the AmI systems; to do this the existing objects' features have to be considered and adequately modeled. Moreover, all the new environment elements have to be modeled, they bi-univocally affect the software design.

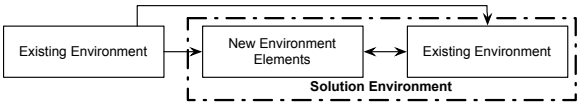

Fig. 1. Relationship among existing and solution environment

This way of conceiving smart environments leads to the metamodel shown in Fig. 2. As we can see, the requirement analysis metamodel grounds on three main domains of an AmI system: *(i)* the *Physical Domain* containing abstractions to model not only the physical composition of the environment with its parts but also the elements acting in the environment and the rules in force; *(ii)* the *Functional Domain* related to the objectives the AmI system must fulfill; *(iii)* the *Organizational Domain* containing elements and relations useful to provide a description of how entities of the smart environment are arranged and interact each other, in various context, playing roles according to specific abilities they own.

The core element of the metamodel is the *Environment* that is composed of *Entities*. We define an *Entity* as something with an independent existence. Thus an entity can act in the environment and can modify the environment configuration. Knowledge about the environment is captured by means of an ontological model that is a logical extension of what we already experienced in PASSI [12] and ASPECS [13]. The ontological modelization of environmental knowledge is based on the primitives defined in their metamodels, that are: *Concept*, *Predicate* and *Action*. A *Concept* is usually used in a broad sense to identify "anything about which something is said"[10]. A *Predicate* is the expression of a property, a state, a constraint or more generally a clarification to specify a *Concept* or a constraint to an *Action*. An *Action* represents "the cause of an event by an acting concept" (adapted from [18]). An *Action* can be intentional or unintentional. The intentionality implies a kind of consciousness to act, capacities to plan and enact strategies for the achievement of a purpose. Therefore, in order to perform an *Intentional Action* the entity should be able to carry out a more or less complex reasoning such as having the ability to acquire and apply knowledge. This means that the entity should be endowed with some kind of intelligence. Conversely, an *Unintentional Action* is an automatic response, the result of fixed rules or a particular set of circumstances, in other words a purely reactive action in the sense of automatics control theory.

In our metamodel, we distinguish entities able to perform Actions (i.e: Smart and Dumb Entities) from entities that are not able to perform actions (i.e: Passive Entities). *Smart Entities* are cognitive and intentional entities. They own knowledge about the world on which they can reason and they have objectives that motivate their actions (intentionality). They are also rational, that is smart entities perform actions according to the principle of rationality expressed by Karl R.Popper [22] and adapted by Newell [19] in the context of knowledge based system. We can rephrased the well known principle as follows: "*If a Smart Entity has knowledge so that one of its actions will lead to one of its goals, then the Smart Entity will select that action*". Summarizing, see Fig.2, Smart Entities have knowledge in form of predicates about the states of the world, they exhibit intentionality because they can perform intentional actions and they plan their actions according to the rationality principle. Moreover, *Smart Entities* are also able to perform unintentional actions. They populate the environment and make it intelligent. They may have or not a physical body. To the first category both *Living Beings* and *Artificial Beings* belong. These latter are artificial entities showing their own physical individuality such as robots. *Artificial Systems* belong to Bodiless Smart Entity cate-

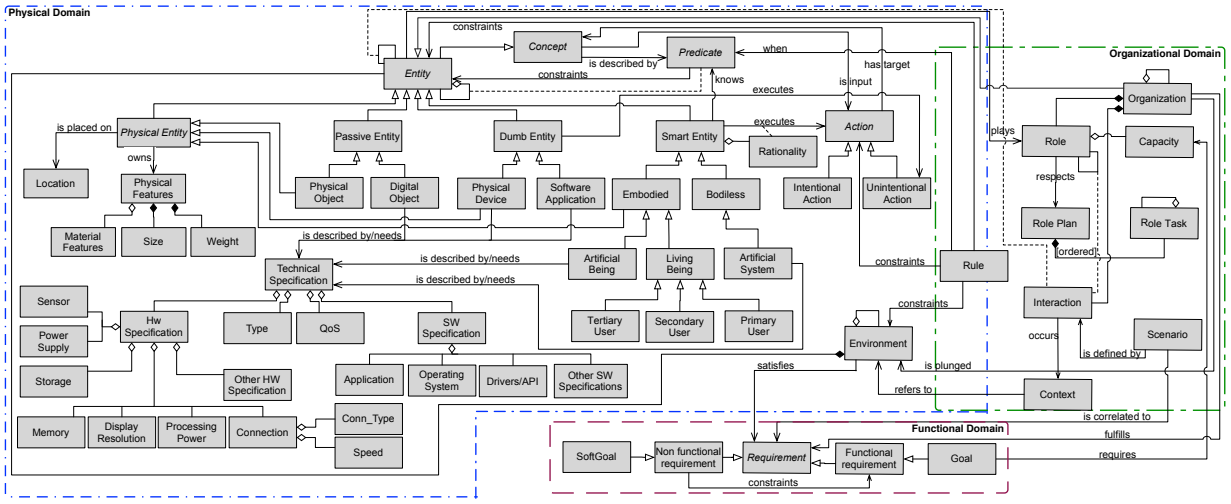

Fig. 2. The Problem Domain Metamodel

gory. They are intelligent software systems (e.g.: Computer Vision systems, Multi agent Systems, Chatbot, etc . . .) that may include hardware parts embedded into the environment. *Living Beings* are specialized in *Primary User, Secondary User* and *Tertiary User* that we adapted from [15]. The *Primary User* is a *Living Being* to which an AmI system is intended for. The AmI system provides functionality and services that satisfy the user requirements. The Primary users are those persons who actually interact with some entities in the environment. The *Secondary User* represents those persons who will occasionally and purposely interact with the AmI system. Finally, *Tertiary User* represents persons that the AmI system is able to perceive and manage. The *Tertiary User* can influence or interact not intentionally with the AmI system that does not provides any service for her/him. *Dumb Entities* are able to perform only *Unintentional Actions*. They are resources without any kind of intelligence, they only react to events or changes in the environment. These elements can be: *(i) Physical Devices*, in other words devices that provide some kind of functionality and that can react executing a mechanic, electromechanical, electronic or software control (e.g.: thermostat, hand dryer, printer, etc . . .); *(ii) Software Applications* that can act in the environment performing some kind of control or can provide services (e.g.: finger recognition, Tv programs recording, etc. . .).

Passive Entities can represent both physical objects composing the environment and digital objects such as electronic documents. Passive Entities are objects of the environment that are not able to perform actions but can be manipulated by means of actions.

Entities having a corporality also belong to the *Physical Entities* category. These latter occupy a position (*Location*) inside an environment and they are characterized by *Physical Features*: *Size*, *Weight* and *Material Features*. Examples of material features are: capacity, fire resistance, conductivity. We would like to point out that not all entities need to be described by means of Material Features, for instance Living Beings. Moreover, except for *Physical Objects* and *Living Beings*, all remaining entities can be described by (or in some case they need of) some *Technical Specifications* in order to establish what services they may provide, how they can be integrated in the environment and what such an entity need for working in a smart environment. Fig.2 shows some common specifications that can be useful for performing a detailed analysis of environment components. Because of space concerns we do not detail these elements.

Entities are commonly subjected to *Rules*. We define a *Rule* as a set of explicit statements or principles governing the functionality, the conduct or the procedures within a particular domain, commonly prescribing what is possible or allowable. A *Rule* represents constraints that limit the behavior, the interactions and the physical configuration of entities. The whole environment can be affected by rules too (e.g.: a laboratory located on the Moon or on the Earth will be affected by a different gravity acceleration). *Actions* can be constrained by *Rules* too.

The smart environment to be implemented has to fulfill different Requirements. The *Requirement* is the key element of the functional domain. We adopt a classical standard definition of Requirement [2]. In particular, *Functional Requirements* describe the functions the software has to execute. They can be statements of services the system should provide, descriptions of how the system should react to particular inputs or how the system should behave in particular situations. *Nonfunctional Requirements* act to constrain the solution. They can be constraints on the services or functions provided by the system such as timing constraints, standards to be compliant to, etc. . *Goals* and *Softgoals* are a specialization of functional and nonfunctional requirements respectively. A *Goal*, representing an actors strategic interest, can satisfy a system requirement. While *Softgoals* [5] are generally considered as goals for which it is difficult to decide whether they are satisfied or not. In our model we use *Softgoals* in order to constrain *Goals*. A *Requirement* is, as usual, correlated to several *Scenarios* that are a description of the interactions occurring in the environment. Therefore, a *Scenario* describes a way in which a system is, or is expected to be, used in a specific context. The *Context* consists in the set of circumstances that form the setting for an event, the surrounding in which interactions occur among entities. These circumstances are expressed in terms of what it can be fully understood and assessed (i.e.: shared knowledge) by the participating entities along with all the elements of their surroundings. Therefore, a *Context* refers to a portion of knowledge related to the environment.

A *Requirement* can be usually satisfied by means of an *Organization* that is the key element of the organizational domain. An *Organization* is defined by a collection of *Roles* that participate to *Interactions* with other roles in a predefined *Context*. An *Organization*, as a whole, is considered an Entity. As well as a single entity, an organization may own an intelligence. We define intelligence the ability of organization to adapt its organizational structure and its dynamics as a response to changes. The organization's intelligence is a consequence of capacities owned by the organization, both by means of smart entities playing roles in it and also as a consequence of the emerging behavior resulting by the interactions among roles, even if they are played by dumb entities. In addition, an *Organization* is governed by *Rules* that regulate what a Role can or not can do and how roles can interact. *Rules* can constraint the organization structure too. A *Role* gathers a set of *Capacities* and a set of rights, obligations and responsibilities. The *Capacity* represents the competences required for realizing some functionality in a specific context independently of the way it is realized.

A *Role* is played by an entity and it describes what an entity should be able to do in order to satisfy requirements. A *Role* respects a *Role Plan* that it composed of *RoleTasks*. The goal of each Role is to contribute to the fulfillment of (a part of) the requirements of the organization within which it is defined. A *RoleTask* defines a part of a role behavior. A Role Task may be atomic or composed by a coordinated sequence of subordinate Role Tasks. The behavior of a Role is specified within a *Role plan*. A *Role plan* is a description of how to combine and to order Role Tasks and interactions to fulfill a (part of a) requirement.

We point out that some of the above definitions are influenced by the ASPECS[13] metamodel. In the next section we provide guidelines for instantiating each element of the physical domain during the requirements analysis activities.

4. Early Requirements Analysis Phase for AmI Systems

In this section we present an Early Requirements Analysis phase specifically conceived for the design of AmI systems in accord to the abstractions we have previously defined. It is composed of two activities named *Environment Physical Description* and *Environment Ontological Description* (see fig.3). We detail each activity describing the purpose, the guidelines for performing the work to be done and the resulting work

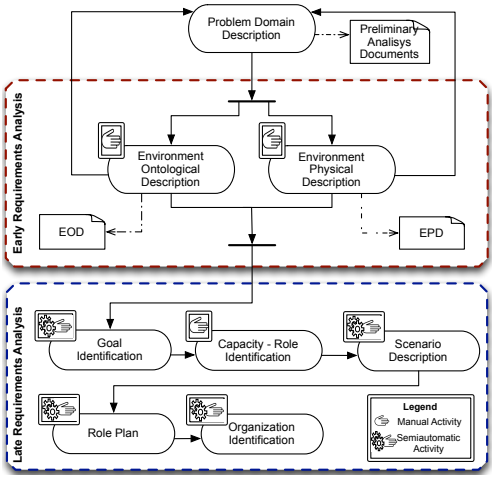

Fig. 3. Activities in the Early and Late Requirements Analysis

products. We assume this phase is preceded by a Problem Domain Description activity that has produced an high-level description of the AmI system to be. Usually, typical scenarios of usage of the AmI system to be are also described. We also assume that interviews have been conducted to cover specific issues, such as: *i)* the overall objectives the AmI system has to reach in order to satisfy business customer expectations; *ii)* the functionality the AmI system has to provide from the user perspective; *iii)* the "smartness", the set of environment elements that have to show intelligent behaviors in the user envisioned scenario; *iv)* finally, the set of constraints or laws (if any) that are in force in the particular domain. We assume, as usual, that this acquired knowledge is gathered in a set of unstructured textual documents that are written in natural language, hereafter named *Preliminary Analysis Documents*.

Environment Physical Description (EPD) - The Environment Physical Description activity aims to model the physical composition of the environment in which the AmI system will be plunged. Here, we represent both the physical configuration of the environment and the technical/normative specifications that constrain it.

- *Guidelines*. Starting from the Preliminary Analysis Documents, the system designer has to collect/produce representations of the environment structure. Thus, (s)he has to identify the static and inactive elements (i.e: walls, roads, pillars, etc...) relevant for the description of the AmI system to be or affecting the next phases of the design process. When appropriate, parts of the environment can be grouped to form sub-environments. For example, a house can be considered as an entire environment or as an aggregation of rooms according to the particular functionality the AmI system will provide. In addition, the designer has to collect technical work products needed to describe infrastructures and components of the environment, such as electrical and hydraulic plants, heating/cooling plant, device technical specifications and so on that could have impact to the design process. Other aspects such as physical laws and/or atmospheric conditions could be considered and described by means of specific technical work products when they are relevant for the problem. The information produced during this activity could reveal deficiencies and/or inconsistencies hidden into the documents coming from the problem domain description. For example, some devices could be not adequate to the technical specifications of existing plants or vice versa. In this case, the designer can iterate the process going back to interview the stakeholders. In order to easy the possibility to integrate components, all restrictions imposed by each of them and by the physical structure of the environment have to be considered and a set of compliance has to be established. The information handled during this task can be captured by means of the metamodel elements concerning the entity features (Physical Features, Location, Technical Specification).

- *Work Products*. We named *Environment Physical Documents* the resulting work product. It is a composite of diagrams, drawings and textual documents that provide details concerning both the physical arrangement of the environment and the technical and normative specifications of the AmI system to be realized.

Environment Ontological Description (EOD) - The Environment Ontological Description activity aims to describe the elements of the environment and their relations. Its final objective is to obtain a deep and unambiguous knowledge of the environment and the problem the system has to address. This activity produces an ontology that specifies all the involved entities, how they act in the environment and how their actions are constrained.

- *Guidelines*. The designer has to perform the ontological description of the environment starting from the Preliminary Analysis Documents. This is done by identifying in these documents the elements (such as entities, actions and predicates according to the metamodel shown in Fig.2) that describe the intelligent environment. To do this, elementary sentences (subject, verb and complements) are extracted from the textual documents and sorted according to semantic similarity. Ambiguous sentences are isolated and not processed. While sentences with the same meaning are discharged and synonyms are identified. Then, nouns are extracted from text creating a list of items grouped in different clusters according to their grammatical function within the sentence. Thus, a noun used as: *(i)* a 'subject' followed by a verb is a candidate Entity if the verb describes an Action on the domain; *(ii)* an 'adjective' is a candidate Predicate of the noun it describes; *(iii)* 'direct object' is a candidate Entity. Secondly, the verbs are identified. In a sentence, a verb may indicate: *(i)* an action performed by the subject of the sentence; *(ii)* a relation among nouns such as aggregation (PART-OF), inheritance (IS-A) or generic association. Finally, the adjectives are identified to be candidate Predicate and can help to identify the nature of the entities (passive, smart, dumb, living being, artificial being, etc. . .). Then, (s)he can establish the kind of the actions (in-

tentional or unintentional) performed by the identified entities. The activity is typically performed in an iterative way to clarify ambiguities (if any) that can emerge from the preliminary documents.

The obtained ontology is refined by adding predicates about states of the world an action produces. Then, his knowledge may be used by smart entities to choose the appropriate action thus acting according to the rationality concept. All these predicates along with the others about elements of the domain contribute to form the whole knowledge base of smart entities.

- *Work Products*. This activity results in a work product named *Environment Ontological Diagram*. Hereafter we assume the designer will also use ontological elements as terms of a specific language.

In the following section, we propose a possible Late Requirement Analysis. It is worth to note that the proposed Early Requirement Analysis may be anchored to well known design methodologies such as ASPECS[13], Tropos [5] or others. This is due to the fact that our Early Requirement Analysis provides the modeling infrastructure for better understanding and modeling the environment and then extracting requirements from the results of this work.

5. A Possible Late Requirements Analysis Phase

The Late Requirement Analysis phase for AmI system we suggest (see Fig.3) is a goal-oriented approach composed of six activities where the motif of the whole workflow is the ontological model of the smart environment. In the following we give a quick description.

Goal Identification (GI) - The purpose of the Goal Identification activity is to identify a list of objectives of the whole AmI system. This activity is strongly based on an ontological representation of the problem domain. In our opinion this kind of representation may provide evidences for the identification of goals that are not explicitly expressed by stakeholders. This because the ontology contains the actions carried out in the domain and the entities that execute them. Actions are typically performed to achieve an aim, whereas smart entities are intentional entities that are committed to specific goals. According to our metamodel shown in Fig.2, entities are able to perform actions that are the way to reach goals. Commonly, these goals are not declared. Thus, the aim of this activity is to make explicit goals deducing them from the performed actions. Moreover, here the designer can obtain also some information about the qualitative aspects (softgoals) related to the system to be developed.

- *Guidelines*. These guidelines coming from [24] and they are based on a list of goal patterns that is possible to identifying in our ontology. We also assume that the ontology model has been drawn. In order to identify the goal list, the analyst has to scan the ontology and to identify each couple action-entities, the description of the identified goal prescribes some reasoning about the mutual position of elements in order to discover dependencies. Indeed, for each action presenting inputs, the analyst has to check if the input is a target of another action, if yes probably there is a dependency between the goals associated to that action. Thus according to the the goal patterns presented in [24], it is necessary to establish to which pattern the identified couple action-entity belongs. Thus, the pattern helps in describing the resulting goal in terms of state of the world to be achieved and other useful information.

- *WorkProducts*. The resulting work product, named *Goal Structural Diagram*, is a composite of a goal diagram (such as a UML class diagram) and textual documents containing information about goal dependency, responsibility and final state of the world the goal reaches.

Capacity-Role Identification (CRI) - The objectives of Capacity-Role Identification activity are first of all to identify capacities and secondly to establish Roles by grouping the previous capacities. The detectable capacities can be physical capabilities, expertise/skills, cognitive faculties, social abilities and knowledge that can be seen as means to be opportunely used for reaching goals.

- *Guidelines*. In order to perform this activity, goals and ontology identified in the previous steps provide useful guidelines to identified capacities. Among all identifiable capacities, we named 'manifested' those arising from the intrinsic nature of the entities populating the environment. The identification of manifested capacities (if any) is quite intuitive. In any case, in order to identify all the necessary capacities, a designer can select a goal and evaluate what are the actions and the manipulated concepts inside the portion of the ontology from which the goal is derived. Therefore, if the fulfillment of the goal imply to manipulate concepts trough actions, this means to define specific capacity useful to access this kind of knowledge and to use resources required by the actions. The second task of this activity regards the Role identifications. A Role

is responsible for reaching or contributing to the fulfillment of goals. Then, the capacities needed to accomplish a goal will be grouped in Roles following specific aspects arising from ontology and according to opportunity criteria. Moreover, the identification of manifested capacities can be a useful guideline in order to define Roles that will be entrusted to the entities. In fact, the set of identified capacities $\{c_i \in C\}$ may contain subsets of capacities manifested MC_{E_j} by specific entities E_j, that is $MC_{E_j} = \{c_i \in C \mid c_i \; is\; manifested\; by\; E_j\}$. Thus, the capacities of subset MC_{E_1} manifested by a specific entity E_1 may be aggregate in Roles to be assigned to E_1.

- *WorkProducts*. This activity results in a Roles Diagram (such as a UML class diagram) containing information about goals the roles are responsible for, the capacities they own, contexts in which they work (i.e: knowledge they need to perform their tasks).

Scenario Description (SD) - The goal of the Scenario Description is to describe the interactions among entities/roles involved in a scenario. A scenario thus describes, in a dynamic way, the flow of actions performed by entities/roles upon other entities/roles.

- *Guidelines*. Starting from the Preliminary Analysis Documents and the Environment Ontological Diagram, the designer has to describe the context in which the interactions occurs (i.e: at work, at home, with friends etc...) along with the related knowledge. Then, (s)he has to explore all the meaningful social interactions among entities/roles. These interactions are reported in a sequence diagram along with all the entities/roles involved in the examined scenario and all the handled concepts.

- *WorkProducts*. The resulting work product, named Scenario Description Diagram (SDD), is a composite of several diagrams (such as UML Sequence Diagrams) and textual documents that provide a dynamic view of the possible scenarios along with a description of relative context in which a scenario occurs.

Role Plan (RP) - This activity allows to describe how a role performs a task in order to reach goals which it is responsible for.

- *Guidelines*. Starting from the interactions described in a previously identified scenario, a designer can specify the tasks a role has to perform for reaching its own goal and in which temporal order. In addition, the ontological description of the problem domain gives information about conditions in which tasks can be performed.

- *WorkProducts*. During this activity the designer has to produce a Role Plan diagram that can be represented by mean of an UML activity diagram.

Organization Identification (OI) - The aim of the Organization Identification activity is to identify organizational structures that show global behaviors finalized to reach some correlated goals, usually according to qualitative aspects.

- *Guidelines*. Starting from Goal Structural and Environment Ontological Diagrams, the designer has to identify groups analyzing portions of ontology under the responsibility of entities. Thus, (s)he has to select the entities that are linked by *Is-a* or *Part-of* relationships. Then, they are grouped according to the Roles they play and goals they have to fulfill.

- *WorkProducts*. This activity results in an Organizational Diagram showing Groups, Roles and relation among Roles.

6. Case Study and Discussions

In this section we present a case study of an AmI System in an industrial context with the aim to prove what we claimed in the section 1. Furthermore, we illustrate and discuss several quality properties showed by our approach.

We are currently working on a project concerning an intelligent logistic warehouse of a supply chain wherein goods are delivered by long distance trucks, processed and then transferred on smaller vehicles for city-scale distribution. This case study allowed us to make experiences about some issues that can arise from designing and developing AmI systems. In the collected scenarios, goods arrive packed in large packs (named pallets) delivered by freight forwarders. The incoming pallets are then processed by several entities such as Automated Guided Vehicles (AGVs), human workers and sorter machines. While the outgoing goods are loaded on eco-trucks and sent toward their final destination. For space concerns, in the following we explain only a portion of the problem.

During our activities we gathered several technical specification documents provided by the stakeholders about the whole environment and its components. Specifically, they provided us a planimetry of the warehouse where areas forbidden to AGVs were highlighted. They also give us the technical documents related to the specific AGV and the sorter machine to be used. We also received specifications about pallets (e.g.: shape, size, weight etc...) and containers (e.g.: height and width of the entrance, pallets disposi-

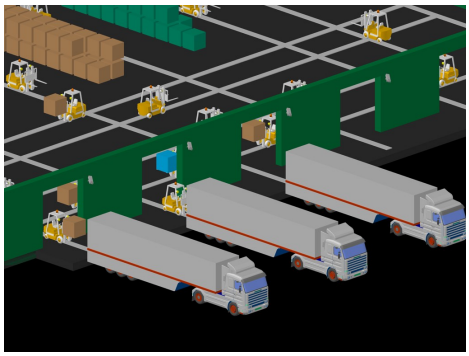

Fig. 4. The Physical composition of the intelligent warehouse

tions etc. . .). We experienced the strong influence the physical configuration of the environment has in the design process. To mention a simple episode, in this study the supplied AGVs were based on optical guidance. They were able to move only by following colored traces painted on the floor. This was a fundamental requirement to identify because all the remaining parts of the AmI system (environment, hardware and software) strongly depended on the configuration of these colored lines. For example, the software solution for AGV navigation was influenced by the lines color and configuration whereas the curves of lines depends on the AGV technical features. We have been able to identify this requirement because we had all the technical documentation necessary available for an accurate analysis of the problem. Hence, the need for refining our design process by adding a new activity for the analysis of physical requirements of an AmI system. This way to perform an early requirements analysis (see EPD guidelines in section 4) allows us to explore constraints, limits and the system feasibility. From Fig.4 we can see a simple but useful graphical representation we produced during EPD activity. This figure shows the envisioned warehouse wherein pre-existing elements are merged with new ones in order to form the physical smart environment to be.

Additionally, during requirement analysis we commonly manage a huge amount of knowledge, often not shared by all the actors participating to the process. It is widely accepted that errors detected during the requirements analysis have less expensive impact on the whole process than an error discovered during the development phase coming from an analysis mistake. The formalization of knowledge and consequently the possibility of validating and sharing it among all participants (customers, designers, analyst, developer, etc.) is a powerful way to avoid misunderstandings and to identify deficiencies in requirements elicitation. The way we suggest to perform the EOD activity (see EOD guidelines in Section 4) by isolating the ambiguities to be processed in a second iteration of the activity and by grouping semantically similar sentences and identifying synonyms allows to disambiguate and improve the understanding of the problem statement. Hence, the requirement elicitation is improved as a consequence.

Fig.5 shows an excerpt of formalized knowledge of the domain concerning our case study. Here terms used to identify instances of entities are shared by the project team. For example, the term *waybill* refers to a digital object representing a list of goods carried by a forwarder inside a pallet. This term derives from the disambiguation process performed to build our ontology. In fact, inside the preliminary analysis documents this object (as well as many others) was called with different names as well as in some cases the same term was used to define different things. Thus, this allowed the team to understand knowledge representations such as it is shown in Fig.5 at a glance. Moreover, both the unification and the separation of knowledge portions according to semantic similarity from one side avoid the redundancy of requirements and from other side allow the identification of new ones.

Moreover, this case study allows us to prove the applicability of our AmI model. We refer to a portion of knowledge of our problem domain represented in Fig. 5 to provide an example. As we can see: *(i)* we succeeded in representing the passive entities that can be both elements of the physical configuration of the environment (e.g: Gate) and elements manipulated by other entities (e.g.: Pallet, Waybill and Billboard); *(ii)* we have been able to identify smart entities such as AGV, Yard Manager and Forwarder belonging to three different categories such as Artificial Being, Artificial System and Living Being respectively. In particular, the AGVs and the Yard Manager are modeled as smart entities inserted as part of the "intelligent warehouse to be" and related, hence plunged, with a pre-existing part of the environment; *(iii)* we have also modeled what entities are able to do in the domain identifying their actions (e.g.: The Yard Manager is able to recognize a Forwarder, to select an appropriate Gate to show on a Billboard and to update the Waybill); *(iv)* we have identified and modeled some rules. In Fig.5, we have modeled three behavioral rules (Permit, Deny and Obligate). When a Pallet is identified, an AGV is obligated to deliver it to the sorter. When a Pallet is not identified, an AGV has a deny to deliver it to the sorter

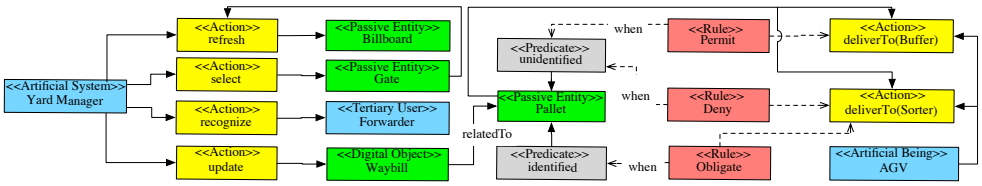

Fig. 5. A portion of ontology for requirements analysis of smart warehouse .

but the Permission to deliver it to a buffer area (that is a specific area of the warehouse). As claimed, we have also identify goals the AmI system has to satisfy and Roles to be played by the environment entities. Following the guidelines provided in [24] goals identified from the portion of knowledge modeled in Fig.5 are: *to refresh Billboard*, *to recognize Forwarder* and *to deliver Pallet*. The first ones derive from the actions performed by the Yard Manager while the latter is related to the AGV actions. Moreover, in this portion of ontology the Role identification is quite simple. According to CRI guidelines (see Section 5), by the analysis of the portion of ontology from which the previously identified goals derive, we have identified capacities such as: ability to access and manage the billboard, knowledge needed to recognize forwarders, ability to load things and transport them to a destination. Thus in our domain, we have identified the role of *Unloader* that include the manifested capacities of an AGV to load and transport packs. This means that in the AmI system to be the role of *Unloader* may be played by the real AGV. At the same time the role of *Unloader* is responsible for achieve the "To deliver Pallet" goal. Another role we identify is the *Incoming Goods Controller* responsible to pursue *to refresh Billboard* and *to recognize Forwarder* goals. This role has to own the ability to access and manage the billboard, and the knowledge in order to recognize forwarders.

6.1. Discussions

We think our approach owns several advantages we have summarized in the following.

Requirement Elicitation Improvement - Ontologies are a possible answer to a question like *what are the classes of entities needed for a complete description and explanation of everything happening in the environment?* The use of an ontological approach helps in deeply understanding the domain in which we are moving. In the field of software engineering, this kind of approach could allow the designer and the stakeholders to refine and disambiguate the problem statement and the application scenarios of the smart environment. Features that can improve the requirements elicitation process are: *(i)* the *completeness* we pursued covering a wide range of abstractions for the Physical Domain conceptualization (see Fig.3); *(ii)* the *accuracy* we researched trying to be reasonably detailed; *(iii)* the *density* of relationships between the represented abstractions; *(iv)* the inherent *expressiveness* of an ontology. Moreover, because we adopt the method defined in [24] for extracting goals from ontology, the more the ontology is refined the more we are able to extract goals. In this way we try to pursue the completeness of requirements.

Environment Comprehension Enhancement - Gathering and analyzing physical features of the existing environment in which the AmI system will be deployed is fundamental. Design choices, in fact, strongly depend on these physical specifications. The introduction of a specific activity during the requirement analysis phase avoids design errors that if late detected may cause a waste of expensive resources.

Awareness - The deeper understanding of the domain required by the construction of an ontological representation of the problem creates a considerable awareness of the problem itself in the designer. From the system perspective, awareness refers to the knowledge about the environment owned by the smart entities. This awareness of the environment is defined by means of the knowledge contained in *Rules* (such as laws, norms, behavioral rules etc. . .) and in the context in which the interaction occurs.

Complexity Reduction - Complexity measures how intricate the environment is in terms of relations and components. An environment seen as a *whole* shows a more complex behavior than an environment seen as distinct parts.We try to reduce the complexity in two ways: (i) by adopting organizational schemes that allow us to analyze the combination among parts and to reason in terms of groups, roles and norms. We see also norms as structural constraints on the system formation that can be seen both as a technical constraint on the system architecture but also as a constraint on the interaction among roles of the entity in the AmI

system; (ii) by decoupling entities (playing roles) that compose the environment in smart and dumb entities. This distinction allows to identify where the smartness of the environment lies and to define a possible risk driven life cycle. In fact, the design of a dumb entity is less costly and risky than a smart entity both in terms of time spent for the design and for the approach and technologies to be used for its development. Conversely, the identification of smart entities allows to define since the early analysis phase the components of AmI system that require complex reasoning mechanisms. In this way, it will be easier to analyze organized and decoupled parts than a single block.

Controlledness - Using the term controlldeness we want to define the freedom degree of the system referred both to technical choices and to the behaviors of the system. The definition of Rules that constraint some aspects of the environment allows the designer to conduct a design process within fixed rails. Therefore, the identification of rules, norms, laws etc... as requirements of the AmI system to be, allows to establish during the designing phase constraints on technological choices to be adopted and system architectures to be used. From the system perspective this means also that in the new environment some entities and behaviors are not allowed. In this way, we define the boundary of the system.

Integrability - An AmI system commonly owns a relevant physical part that needs to be analyzed in order to avoid integrability issues among the different components it will be composed of. This issue requires more attention in case the system has to be deployed in a preexisting environment where the already existing devices, infrastructures and physical configuration, impose constraint on their merging. Examples of possible constraints can be incompatibility of communication protocol among devices, presence of reinforced concrete walls that prevent wireless transmissions, electrical plant inadequate. In the development of an AmI system, a requirement is also to avoid costly changes when an alternative solution can be considered. For this reason, we proposed the Physical Environment Description activity during the early analysis phase. During this activity, the designer collects or produces all documents needed for understanding both the preexisting physical environment (if any) and the devices to be adopted and (s)he determines the set of rules to be comply.

Intentionality - Another key concept of our approach lies on the 'intentionality' we attribute to smart entities. Several studies try to understand what are the metrics to measure the intelligence of a system. In this work we talk about intelligence in the narrower sense of 'smartness'. By introducing the concept of intentionality in our requirements metamodel we have given both a further feature in order to identify intelligent entities but also the possibility to identify goals starting from the ontological description of the domain.

Reuse and Automatization - The use of an ontological approach in order to acquire and formalize domain knowledge is also, in our envisioned complete methodology, a way to be able to reuse portions of knowledge and to automatize some parts of the methodological approach. As it concerns the reuse, in our complete design process we are working on, portions of knowledge of the problem domain have a univocal relation with elements of the solution domain. Thus, if a new system in the same domain of a previous one is going to be designed, it will be possible to find matching of common knowledge and consequently portions of solution to be reused. As it concerns the automatization, it is well known that an ontology is a machine-readable representation of knowledge. Because some parts of our design process are strongly based on approaches and algorithms that infer elements from the ontology, we are developing a tool that automatizes these algorithms in order to reduce the manual work of a designer.

Organizational and Normative Principles - The introduction in our metamodel (see Fig.2) of abstractions such as roles, organizations and rules (in this case in the sense of norms) provides the possibility to model more complex organizational patterns such as holonic ones in the envisioned world of smart*.

Weakness - Our approach allows to transform knowledge from an unstructured form to a structured one. This could result in an extra effort in the case of small size problems where it could be more easy to directly identify goals from domain description, user interviews, etc. Moreover our approach is tied to be implemented in a platform that supports normative and organizational principles. In the same way, our analysis process is conceived to provide advantages for the development of agent systems based on BDI paradigm. Thus in the case the implementation platform does not support norms, organizations and BDI agents, the effort to be spent for applying our approach may not be justified.

7. Conclusions and Future Works

The study of ad hoc methodologies able to address specific concerns about AmI system design is currently

an open challenge. In this work we contribute to it by proposing some useful abstractions along with two activities for performing an early requirements analysis of AmI systems. We also suggest a possible late requirements analysis and underline that the activity we propose can be anchored to well-know metodologies. We defined metamodeling elements that can be the bridge between our early requirements analysis and other methodologies. Moreover, we would like to point out we are completing an ontology-based methodological approach where we use the ontology not only as a descriptive model but also as a mean for deducing system goals and for building the agent knowledge base. In this paper we have shown only the use of the ontology during the early requirement phase because we are addressing the others two aspects in specific works. We are finalizing a complete methodological approach able to cover the design of AmI system from the analysis phase to the development one and a tool for automatizing and supporting the process.

References

[1] Philips research. ambient intelligence: Changing lives for the better. www.research.philips.com, 2007.

[2] A. Abran, J.W. Moore, P. Bourque, R. Dupuis, and L.L. Tripp. *SWEBOK®: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.

[3] Uwe Aßmann, Steffen Zschaler, and Gerd Wagner. Ontologies, meta-models, and the model-driven paradigm. *Ontologies for Software Engineering and Software Technology*, pages 249–273, 2006.

[4] Juan Carlos Augusto. Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. In *Intelligent Computing Everywhere*, pages 213–234. Springer, 2007.

[5] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[6] Kevin Brooks. The context quintet: narrative elements applied to context awareness. In *Human Computer Interaction International Proceedings*, volume 2003, 2003.

[7] Verónica Castañeda, Luciana Ballejos, Ma Laura Caliusco, and Ma Rosa Galli. The use of ontologies in requirements engineering. *Global Journal of Researches In Engineering*, 10(6), 2010.

[8] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277 – 298, 2009.

[9] Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009.

[10] O. Corcho and A. Gómez-Pérez. A roadmap to ontology specification languages. *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pages 80–96, 2000.

[11] Antonio Coronato and Giuseppe De Pietro. Formal design of ambient intelligence applications. *Computer*, 43(12):60–68, 2010.

[12] M. Cossentino. From requirements to code with the PASSI methodology. In *Agent Oriented Methodologies*, chapter IV, pages 79–106.

[13] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam. ASPECS: an agent-oriented software process for engineering complex systems. *Autonomous Agents and Multi-Agent Systems*, 20(2):260–304, 2010.

[14] J. Encarnacao E. Aarts. *True Visions: The Emergence of Ambient Intelligence*. Springer, 2006.

[15] Kenneth D Eason. *Information technology and organisational change*. CRC Press, 1989.

[16] J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley, 1999.

[17] J. J. Gomez-Sanz, J.M. Fernandez-de Alba, and Fuentes-Fernandez R. Ambient intelligence with ingenias. In Jorg P.Muller and Massimo Cossentino, editors, *The 13th International Workshop on Agent-Oriented Software Engineering*, pages 141 –158, 2012.

[18] E.J. Lowe. *A survey of metaphysics*. Oxford University Press Oxford, 2002.

[19] Allen Newell. The knowledge level. *Artificial intelligence*, 18(1):87–127, 1982.

[20] L.S. Passos, R.J.F. Rossetti, and J. Gabriel. An agent methodology for processes, the environment, and services. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 2124–2129, 2011.

[21] Juan Pavòn, Jorge J. Gòmez-Sanz, and Rubén Fuentes. The INGENIAS methodology and tools. In *Agent Oriented Methodologies*, chapter IX, pages 236–276. Idea Group Publishing, 2005.

[22] Karl Popper. The myth of the framework. In *Rational Changes in Science*, pages 35–62. Springer, 1987.

[23] Carlos Ramos, Juan Carlos Augusto, and Daniel Shapiro. Ambient intelligence—the next step for artificial intelligence. *Intelligent Systems, IEEE*, 23(2):15–18, 2008.

[24] P. Ribino, M. Cossentino, C. Lodato, S. Lopes, L. Sabatucci, and V. Seidita. Ontology and goal model in designing bdi multi-agent systems. In *Proceedings of the 14th Workshop "From Objects to Agents" WOA 2013*, volume 1099, pages 66–72. http://ceur-ws.org/Vol-1099/, Torino, Italy, December 2-3, 2013.

[25] Fariba Sadri. Ambient intelligence: A survey. *ACM Computing Surveys (CSUR)*, 43(4):36, 2011.

[26] M. Shibaoka, H. Kaiya, and M. Saeki. GOORE: Goal-oriented and ontology driven requirements elicitation method. In *Advances in Conceptual Modeling–Foundations and Applications*, pages 225–234. Springer, 2007.

[27] K. Siegemund, E. J Thomas, Y. Zhao, J. Pan, and U. Assmann. Towards ontology-driven requirements engineering. In *Workshop Semantic Web Enabled Software Engineering at 10th International Semantic Web Conference (ISWC), Bonn*, 2011.

[28] Danny Weyns. An architecture-centric approach for software engineering with situated multi-agent systems. volume 22 of *The Knowledge Engineering Review*, pages 405–413. Cambridge University Press, Dec-2007.