Chapter 3

AN ARCHITECTURE THAT SUPPORTS TASK-CENTERED ADAPTATION IN INTELLIGENT ENVIRONMENTS

Achilles D. Kameas

DAISy group, The Computer Technology Institute, University of Patras Campus Patras, Hellas

School of Sciences and Technology, The Hellenic Open University, Patras, Hellas

kameas@cti.gr

Christos Goumopoulos

DAISy group, The Computer Technology Institute, University of Patras Campus Patras, Hellas

goumop@cti.gr

Hani Hagras, Victor Callaghan

The Computational Intelligence Centre, Department of Computing and Electronic Systems, University of Essex, Wivenhoe Park, Colchester, UK

{hani,vic}@essex.ac.uk

Tobias Heinroth

Institute of Information Technology, Ulm University, Ulm, Germany

tobias.heinroth@uni-ulm.de

Michael Weber

Institute of Media Informatics, Ulm University, Ulm, Germany

michael.weber@uni-ulm.de

W. Minker et al. (eds.), *Advanced Intelligent Environments*,
DOI 10.1007/978-0-387-76485-6_3,

Abstract The realization of the vision of ambient intelligence requires developments both at infrastructure and application levels. As a consequence of the former, physical spaces are turned into intelligent AmI environments, which offer not only services such as sensing, digital storage, computing, and networking but also optimization, data fusion, and adaptation. However, despite the large capabilities of AmI environments, people's interaction with their environment will not cease to be goal-oriented and task-centric. In this chapter, we use the notions of ambient ecology to describe the resources of an AmI environment and activity spheres to describe the specific ambient ecology resources, data and knowledge required to support a user in realizing a specific goal. In order to achieve task-based collaboration among the heterogeneous members of an ambient ecology, first one has to deal with this heterogeneity, while at the same time achieving independence between a task description and its respective realization within a specific AmI environment. Successful execution of tasks depends on the quality of interactions among artifacts and among people and artifacts, as well as on the efficiency of adaptation mechanisms. The formation of a system that realizes adaptive activity spheres is supported by a service-oriented architecture, which uses intelligent agents to support adaptive planning, task realization and enhanced human–machine interaction, ontologies to represent knowledge and ontology alignment mechanisms to achieve adaptation and device independence. The proposed system supports adaptation at different levels, such as the changing configuration of the ambient ecology, the realization of the same activity sphere in different AmI environments, the realization of tasks in different contexts, and the interaction between the system and the user.

Keywords: Ambient intelligence; Pervasive adaptation; System architecture; Ambient ecology; Activity sphere; Ontology; Ontology alignment; Agents; Fuzzy agents; Interaction; Interaction modality; Pro-active dialogue.

1. Introduction

Ambient intelligence (AmI) is a new paradigm that puts forward the criteria for the design of the next generation of intelligent environments (Remagnino and Foresti, 2005). The realization of the vision of ambient intelligence requires developments both at infrastructure and application levels. As a consequence of the former, physical spaces are turned into intelligent environments, which offer not only services such as sensing, digital storage, computing, and networking, but also optimization, data fusion, and adaptation. Intelligent computation will be invisibly embedded into our everyday environments through a pervasive transparent infrastructure (consisting of a multitude of sensors, actuators, processors, and networks) which is capable of recognizing, responding, and adapting to individuals in a seamless and unobtrusive way (Ducatel et al., 2001). Such a system should also provide the intelligent "presence" as it be able to recognize the users and can autonomously program itself in a non-intrusive manner to satisfy their needs and preferences

(Doctor et al., 2005). AmI offers great opportunities for an enormous number of applications in domains such as health care, the efficient use of energy resources, public buildings, and in leisure and entertainment. Ubiquitous computing applications constitute orchestrations of services offered both by the environment and the information devices therein (Kameas et al., 2003).

Every new technological paradigm is manifested with the "objects" that realize it. In the case of AmI, these may be physical or digital artifacts. The former, also known as information devices, are new or improved versions of existing physical objects, which embed information and communication technology (ICT) components (i.e., sensors, actuators, processor, memory, wireless communication modules) and can receive, store, process, and transmit information. The latter are software applications that run on computers or computationally enabled devices (i.e., digital clocks, MP3 players, weather forecasts etc). Thus, in the forthcoming AmI environments, artifacts will have a dual self: they are objects with physical properties and they have a digital counterpart accessible through a network (Kameas et al., 2005). We shall use the term ambient ecology to refer to a collection of such artifacts that can collaborate to achieve a given task.

An important characteristic of AmI environments is the merging of physical and digital space (i.e., tangible objects and physical environments are acquiring a digital representation); nevertheless, people's interaction with their environment will not cease to be goal-oriented and task-centric. However, we expect that ubiquitous computing technology will allow people to carry out new tasks, as well as old tasks in new and better ways. People will realize their tasks using the services offered by ambient ecologies. Knowledge will exist both in people's heads (in the form of upgraded skills), the ambient ecology and the AmI environment (in the knowledge bases of the artifacts). In most cases, successful realization of tasks will require the knowledge-based adaptation of task models in the changing context, because it depends on the quality of interactions among artifacts and among people and artifacts, as well as on the efficiency of adaptation mechanisms.

Adaptation is a relationship between a system and its environment where change is provoked to facilitate the survival of the system in the environment. Biological systems exhibit different types of adaptation. They have inspired the development of adaptive software systems, which use a mechanism similar to biological ontogenetic adaptation so as to regulate themselves and change their structure as they interact with the environment. This mechanism is based on the replacement of one component by another component, where both components share a common interface. This approach is common to autonomic systems.

In this chapter, we use the notion of activity spheres to describe the specific ambient ecology resources, data and knowledge required to support a user in realizing a specific goal. Activity spheres are discussed in Section 2. The formation of a system that realizes such activity spheres is supported by a service-oriented architecture, which is presented in Section 3. In order to achieve task-based collaboration among the heterogeneous members of an ambient ecology, first one has to deal with this heterogeneity, while at the same time achieving independence between a task description and its respective realization within a specific AmI environment. To this end, we employ ontology alignment mechanisms described in Section 4.

Our system supports adaptation at different levels. At the ambient ecology level, the system supports the realization of the same activity sphere in different AmI environments. At the same time, it will adapt to changes in the configuration of the ecology (i.e., a new device joining, a device going out of service etc). At the task level, the system realizes the tasks that lead to the achievement of user goals using the resources of the activity sphere. Another dimension of adaptation is the interaction between the system and the user. An intelligent (speech) dialogue is able to provide easily understood metaphors for allowing people to tailor and configure ambient ecologies in a semi-tacit way to their needs and mechanisms that allow the man–machine interaction to adapt to the user context and behavior. In order to achieve this, we use a set of intelligent agents to support adaptive planning, task realization, and enhanced human–machine interaction. These are discussed in Sections 5 and 6. In the following subsection, we shall firstly present a scenario illustrating the concepts and mechanisms discussed in the remainder of the chapter.

1.1 Life in Intelligent Adaptive Homes

The scenario that follows is based on the imaginary life of a user (Suki) who just moved to a home that is characterized by being intelligent and adaptive. The scenario will help to illustrate the adaptation concepts presented in the chapter. Figure 1 illustrates the AmI environment described in the scenario.

Suki has been living in this new adaptive home for the past 10 months. Suki's living room has embedded in the walls and ceiling a number of sensors reading inside temperature and brightness; more sensors of these types are embedded in the outside wall of the house. A touch screen mounted near the room entrance together with a microphone and speaker is used as the main control point. Suki can use multiple modalities in order to interact with his smart home. The most powerful ones are the touch screen and the speech dialogue system (SDS): The touch screen can display the situation of the house, the settings, and the commands Suki has given, as well as the rules inferred by the various agents. With the help of the SDS Suki can, for example, voice control all devices and services registered to the sphere and the sphere itself can pro-actively ask Suki for information needed, e.g., to make

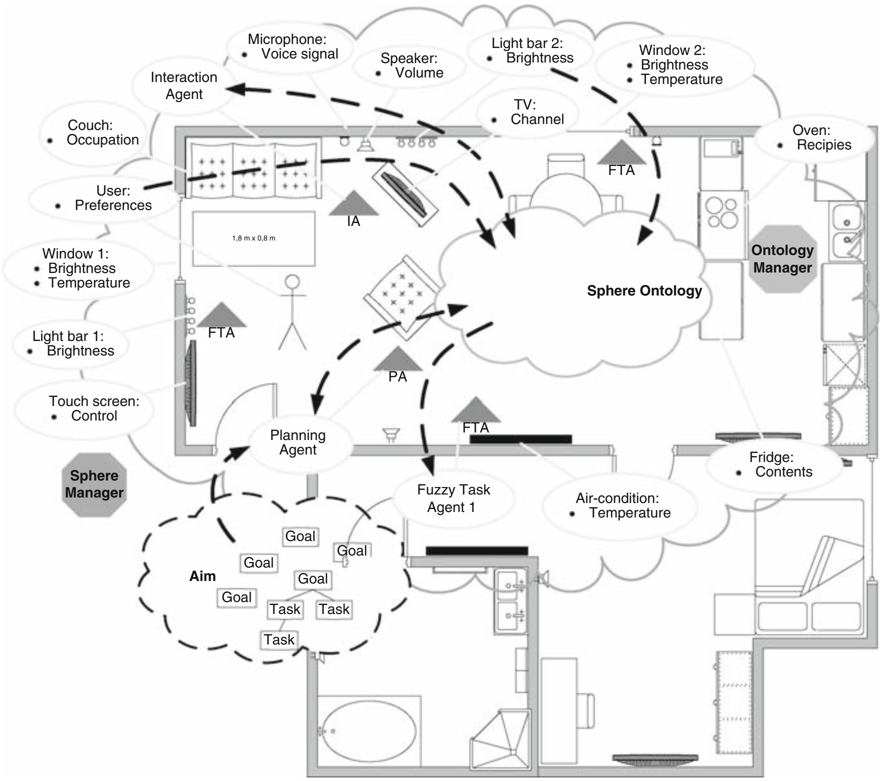

Figure 1. Elements of an activity sphere.

a system-relevant decision. The usage of these devices and services will be described in Section 6.

Suki uses an air-conditioning as the main heating/cooling device. The windows are equipped with automated blinds, which can be turned in order to dim or brighten the room. Also, the windows can open or close in order to adjust the room temperature. For the same purpose Suki can use two sets of lights in the living room. Finally, Suki has two TV sets in the house, one in the living room and one in the kitchen. The latter also contains a smart fridge, which can keep track of its contents, and an oven, which also stores an inventory of recipes and can display them in the fridge screen or the TV set. Each of these devices of the ambient ecology contains its own local ontology, which describes the device physical properties and digital services. For example, the lamp ontology stores the brand, the material, the size, as well as the location, the state (on/off) and the luminosity level. Similarly, the TV set ontology stores the set and screen dimensions, location, state, available TV channels, currently playing TV channel, statistics about channel usage, as well as viewing parameters (brightness, volume, contrast, etc.). The local ontologies will be used to form the sphere ontology, as will be described in Section 4. Moreover, these services can be directly manipulated by the task agents, as will be described in Section 5.

Suki's goal is to feel comfortable in his living room, no matter what the season or the outside weather conditions are. After careful thinking, he concluded that for him comfort involved the adjustment of temperature and brightness, the selection of his favorite TV channel, and the adjustment of volume level, depending on the TV programme.

Regarding the latter, the smart home system had observed Suki's choices over the past months and has drawn the conclusion that he tends to increase the volume when music or English-speaking movies are shown, except when it is late at night; he keeps the volume low when movies have subtitles or when guests are around. This has been possible with the help of the task agent. Nevertheless, the system does not have enough data to deduce Suki's favorite lighting and temperature conditions as the seasons change. Initially, the system will combine information in Suki's personal profile, the environmental conditions, the weather forecast, and anything else that may matter, in order to tacitly adapt to the values that Suki might want. In case of a doubt, it will engage in dialogue with Suki about specific conditions, with the help of the interaction agent. Of course, Suki can always set directly the values he desires by manipulating the devices that affect them; the system will monitor such activity and tacitly will adjust its rules. Dialogue modalities are described in Section 6.

2. Ambient Ecologies and Activity Spheres

For the intelligent ambient adaptive systems, we will introduce the ambient ecology metaphor to conceptualize a space populated by connected devices and services that are interrelated with each other, the environment, and the people, supporting the users' everyday activities in a meaningful way (Goumopoulos and Kameas, 2008). Everyday appliances, devices, and context-aware artifacts are part of ambient ecologies. A context-aware artifact uses sensors to perceive the context of humans or other artifacts and sensibly respond to it. Adding context awareness to artifacts can increase their usability and enable new user interaction and experiences.

An ambient ecology can be composed of individual artifacts and in parallel itself can be used as a building block of larger and more complex systems. Compose-ability can give rise to new collective functionality as a result of a dynamically changing number of well-defined interactions among artifacts. Compose-ability thus helps resolving both scalability and adaptability issues of ambient ecologies.

In order to model the way everyday activities are carried out within an AmI environment populated with an ambient ecology, we introduce the notion of activity sphere (Zaharakis and Kameas, 2008). An activity sphere is intentionally created by an actor (human or agent) in order to support the realization of a specific goal. The sphere is deployed over an AmI environment and uses its resources and those of the ecology (artifacts, networks, services, etc.). The goal is described as a set of interrelated tasks; the sphere contains models of these tasks and their interaction. These models can be considered as the counterparts of programs, only that they are not explicitly programmed, but are usually learnt by the system through observation of task execution. The sphere can also form and use a model of its context of deployment (the AmI environment), in the sense that it discovers the services offered by the infrastructure and the contained objects. The sphere instantiates the task models within the specific context composed by the capabilities and services of the container AmI environment and its contained artifacts. In this way, it supports the realization of concrete tasks.

Thus, a sphere is considered as a distributed yet integrated system that is formed on demand to support people's activities. An activity sphere is realized as a composition of configurations between the artifacts and the provided services into the AmI environment. People inhabit the AmI environment and intentionally form spheres by using the artifacts and the provided services. An activity sphere continuously "observes" people interactions with artifacts in different contexts, can learn their interests and habits, and can exhibit cognitive functions, such as goal-directed behavior, adaptation, and learning.

In the example we provided above, in order to satisfy the goal "feel comfortable", an activity sphere will be set up, as described in the next section. This goal can be described, for example, with abstract tasks as follows:

1 Set a comfortable temperature (TEMP)
  - Sense the indoor and outdoor temperatures;
  - Adjust room heating/cooling according to the user preferences and context.

2 Set a comfortable level of lighting (LIGHT)
  - Sense the indoor light levels;
  - Adjust indoor light levels according to the user preferences and context.

3 Select favorite TV program (FAVTV)
  - Check media options;
  - Set media according to the user preference and context.

The configuration of a sphere could be realized in three ways: explicit, tacit, and semi-tacit (Seremeti and Kameas, 2008). In the former mode, people configure spheres by explicitly composing artifact affordances, based on the visualized descriptions of the artifact properties, capabilities, and services. To operate this mode, people must form explicit task models and translate them into artifact affordances; then they must somehow select or indicate the artifacts that bear these affordances. The independence between object and service is maintained, although there do not exist clear guidelines regarding the degree of visibility (of system properties and seams) that a sphere should offer to people. The tacit mode operates completely transparently to the user and is based on the system observing user's interactions with the sphere and actions within the sphere. In an ideal AmI space, people will still use the objects in their environment to carry out their tasks. Agents in the intelligent environment can monitor user actions and record, store, and process information about them. Then, they can deduce user goals or habits and pro-actively support people's activities within the sphere (i.e., by making the required services available, by optimizing use of resources, etc). The sphere can learn user preferences and adapt to them, as it can adapt to the configuration of any new AmI space that the user enters. To achieve this, the encoding of task- and context-related metadata is required, as well as of the adaptation policies, which will be used by the task realization mechanisms.

The semi-tacit mode realizes a third way of configuring the sphere by combining the explicit and the implicit way. The user interacts, for example, by the use of speech dialogues with the system and provides only basic information regarding his/her goals and objectives. The user does not have to explicitly indicate artifacts and form task models but provides general commands and tells AmI some requirements he has. We assume that the semi-tacit mode may perform better than the tacit mode because the system operation within the AmI space uses the combined outcome of observation and explicit (but more general) user input. Thus it does not operate completely transparently. We assume on the one hand this is more comfortable for the user and on the other hand the system's decisions may be made closer to the user's ideas and even with a more reasonable speed.

3. System Architecture

The system we propose operates in an AmI environment, which is populated with an ambient ecology of devices, services, and people. Our basic assumption is that these components are all autonomous, in the sense that (a) they have internal models (ontologies) of their properties, capabilities, goals, and functions and (b) these models are proprietary and "closed", that is, they are not expressed in some standardized format. Nevertheless, each component can be queried and will respond based on its ontology. Finally, the AmI environment provides a registry of these components, where, for each component, its ID and address are stored.

At the same time, people have goals, which they attempt to attain by realizing a hierarchy of interrelated tasks. These are expressed in a task model, which is used as a blueprint to realize an activity sphere. Thus, for each user goal, an activity sphere is initialized, based on its task model, which consists of all software, services, and other resources necessary to support the user in achieving the goal. A sphere consists of the following (Figure 1):

1 Devices/services/AmI space properties and resources.

2 Goal and task models.

3 Software modules: Sphere manager, ontology manager.

4 Sphere ontology.

5 User(s) and user profile(s).

6 Agents: Planning agent, fuzzy systems-based task agent, interaction agent, device, or other agents (all with their local knowledge bases or ontologies).

A brief description of the components of a sphere follows.

Sphere manager (SM): The SM forms or dissolves an activity sphere and manages its several instances on different AmI spaces. It subscribes to the AmI space registry and operates as an event service for the other system components. The SM is responsible for initializing the other system components (i.e., fuzzy systems-based task(s) agent, ontology manager, etc) and thus interacts with all system components. An important role of the SM is to oversee the fulfillment of the sphere's goal. This is done in cooperation with the ontology manager which provides reasoning services. The SM also provides context-based adaptation of the activity sphere in an AmI space by deciding, for example, the replacement of a device because of a user location change that affects the task operation. The SM could be viewed as the "operating system" of a sphere virtual machine.

Ontology manager (OM): The OM aligns and merges local device, agent, policy, and user ontologies according to the task model that realizes the sphere goal. The OM is responsible for creating, dissolving, and generally managing the sphere ontology and for responding to queries regarding the sphere ontology. To that end, the OM maintains rules and provides inference services. The OM interacts with all system components.

Fuzzy systems-based task agent (FTA): One or more FTA (depending on the goal complexity) oversee the realization of given tasks within a given AmI space. These agents are able to learn the user behavior and model it by monitoring the user actions. The agents then create fuzzy-based linguistic models which could be evolved and adapted online in a life-learning mode. The FTA maintains its own local knowledge base, which is initially formed by the SM, based on the task model and the sphere ontology. New rules generated are exported to the sphere ontology through the OM. The FTA not only interacts with the SM and the interaction agent but also with the devices and services in the AmI space.

Interaction agent (IA): The IA provides a multimodal front end to the user. Depending on the sphere ontology it optimizes task-related dialogue for the specific situation and user. The IA may be triggered by both the FTA and the planning agent to retrieve further context information needed to realize and plan tasks by interacting with the user.

Planning agent (PA): The PA ensures that all tasks in the task model are described in a concrete and realizable manner and that their realization is feasible given time and resource constraints. It interacts with the SM, the OM, and the IA to enable the sphere to deal with near real-time, user-centered planning. It provides plan repairing in case a context-based adaptation by SM cannot be performed.

Sphere ontology (SO): Contains all the knowledge and data that pertain to the sphere. It is the glue that keeps these components functioning together as long as the purpose of the sphere lasts (then it is dissolved). All component interactions take place via or with the help of the ontology. It is formed on demand first by matching and then by merging or aligning the following constituent ontologies (Figure 2):

1 User profile ontologies.

2 Policy ontologies.

3 Agent local ontologies.

4 Device/service proprietary ontologies.

These ontologies are more analytically described in the following:

Device/service ontologies: They are local to each device/service. They are proprietary. We assume they do not follow a standard structure or adhere to a general formal ontology (GFO) or upper ontology. They are maintained by device/service and can be queried in a standard way.

FTA ontology: It is isomorphic to the FTA knowledge base and is maintained by the agent. Initially, it is formed by querying the SO on the devices and services that offer the services necessary for realizing the tasks in the task model. When the knowledge base changes, the agent updates the SO.

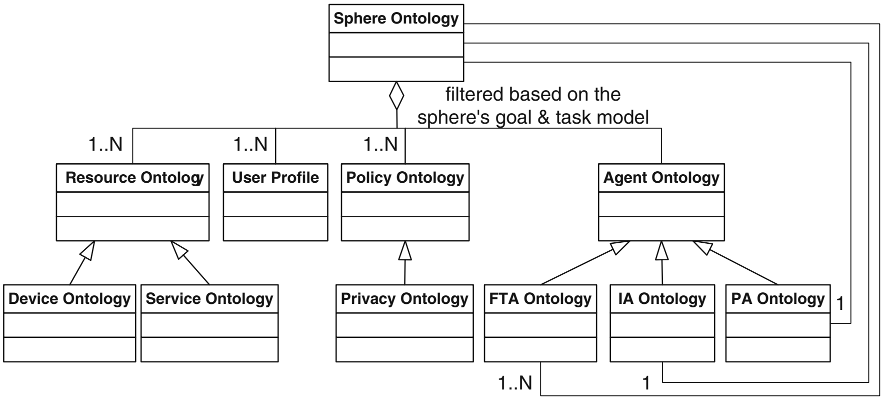

Figure 2. Domain model of sphere ontology.

IA ontology: It is maintained by the IA. Initially, it is formed by querying the SO on the devices and services that offer interaction services, based on the agent interaction policy and the tasks in the task model.

Policy ontologies: They encode entities and rules that describe specific policies, such as user privacy, multi-user conflict resolution, social intelligence, goal fulfillment conditions, and ontology dissolution. They are considered as part of the infrastructure.

User profiles: They encode user traits and preferences. A user can assume different personas based on context. A user profile can be created/updated by the FTA after monitoring device/service usage.

We assume an architecture that is service-oriented and enforces a clean service-oriented design approach, with a clear distinction between interfaces and implementation. This is similar to assumptions of component-oriented software development, as a result of which many application component platforms seamlessly incorporate service-oriented features.

When a new activity sphere is formed, the SM initializes all required components. The following steps are followed:

1 Download goal and task models from library or user profile.

2 Search for AmI space registry.

3 Instantiate PA, which tries to resolve any abstract task descriptions using the registry.

4 Initialize OM, which forms the SO based on the concrete task model and the registry.

5 Instantiate FTA and create an initial rule set using the SO.

6 Instantiate IA and create its local ontology based on interaction policy and SO.

A most common event during the operation of an activity sphere is a change in the registry of the AmI space. This may happen as a consequence of a device or service becoming unavailable, a new device arriving, or even having to instantiate the activity sphere in a new AmI space. In order to adapt to such an event, the system operates as follows:

1 The SM continuously polls for these kinds of changes and when one happens, it creates an event.

2 The PA recognizes the event and recalculates the task model.

3 The OM recognizes the event and starts a new ontology alignment process.

4 The FTA knowledge base is updated, thus the agent can now access the new configuration of the ecology.

5 The IA local ontology is updated regarding changes in the devices offering interaction capabilities.

Returning to the example, the abstract task TEMP could be made concrete by the PA with the use of devices that have been discovered in the ambient ecology. Then, the concrete task may look like the following:

1 FTA senses indoor temperature using Sensor S1 in the living room.

2 FTA senses indoor temperature using Sensor S2 in the bedroom.

3 FTA senses outdoor temperature using Sensor S3.

4 FTA checks Suki's temperature preferences stored in his local ontology.

5 FTA deduces Suki's favorite temperature for the several rooms.

6 FTA adjusts temperature by using the windows.

7 FTA adjusts temperature by using the radiator.

8 IA provides control dialogues (using different modalities) for allowing Suki to directly adjust the temperature.

Based on this description, the ontology manager will create the sphere ontology, as described in the next section, and will create the first version of the FTA knowledge base. Then, the FTA will assume direct control over the devices, monitor their usage, and update its knowledge base and its local ontology.

4. Using Ontologies to Support Adaptation

When realizing ambient spheres, one faces the challenge to develop mechanisms to support the communication between the heterogeneous devices, so as to facilitate the realization of the user's goal supported by the sphere.

In the case of activity spheres, there are multiple causes of heterogeneity:

- artifacts (devices or services) are expected to come with a proprietary, usually closed, model of itself and the world;
- intelligent environments will have their own models of their resources and services;
- user goal and task models as well as policies (i.e., for interaction, privacy, etc) will be expressed in various domain-dependent notations;
- networking and content exchange protocols usually have a restricted closed world model.

One can expect that each device in the ambient ecology will contain at least a description of its services using some popular protocol (i.e., UPnP), or even more, a set of meta-data describing its properties and services. Thus, by manipulating these local "ontologies", one can deal with the problem of heterogeneity, as well as with the problem of representing state changes.

An ontology is usually defined as "a formal, explicit specification of a shared conceptualization" (Gruber, 1993). A "conceptualization" refers to an abstract model of some phenomenon in the world, which identifies the relevant concepts of that phenomenon. "Explicit" means that the type of concepts used and the constraints on their use are explicitly defined. "Formal" refers to the fact that the ontology should be machine readable. "Shared" reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group. Thus, an ontology is a structure of knowledge, used as a means of knowledge sharing within a community of heterogeneous entities.

Currently, there are two major standardization efforts under way in the ontology domain, carried out by IEEE and the World Wide Web Consortium. The former is concerned with a standard for upper ontology, and due to its general approach is likely to have only a limited impact. The proposal of W3C and its ontology task group resulted in the ontology language OWL (Web Ontology Language), which is the evolution of DAML+OIL. The OWL language provides support for merging of ontologies, through the use of language features which enable importing other ontologies and enable expression of conceptual equivalence and disjunction. This encourages the separate ontology development, refinement, and re-use.

The issue we face when building an activity sphere is more complex. Although common ontologies can serve as the means to achieve efficient communication between heterogeneous artifacts, it seems that they are not always effective (i.e., using a common ontology is not possible in the case where artifacts use closed proprietary ontologies). A different ontology-based mechanism is required, which will make the ontologies of the interacting artifacts semantically interoperable.

Ontology matching is the process of finding relationships or correspondences between entities of two different ontologies. Its output is a set of correspondences between two ontologies, that is, relations that hold between entities of different ontologies, according to a particular algorithm or individual. Current techniques for ontology matching require access to the internal structure of constituent ontologies, which must be verified for consistency, and result in static solutions (a set of mappings or a new ontology), which have to be stored somewhere. But an activity sphere is a transitory, dynamically evolving entity, composed of heterogeneous, independent, usually third-party components. That is why we are applying the ontology alignment technique. According to Euzenat and Schvaiko (2007), the ontology alignment process is described as follows: given two ontologies, each describing a set of discrete entities (which can be classes, properties, rules, predicates, or even formulas), find the correspondences, e.g., equivalences or subsumptions, holding between these entities. Based on these alignments, one can apply ontology merging in order to produce the top-level sphere ontology, which realizes an activity sphere.

In the example, based on the concrete task plan, which details the entities that must be used in order to realize an abstract task, the ontology manager forms the sphere ontology, which contains information about the following:

- the states of the devices and services that participate in the task;
- the knowledge bases of the sphere agents;
- the user profile;
- the constraints and policies that apply to the realization of the goal and its tasks.

5. Realizing Adaptation Over Long Time Intervals with the Help of a Fuzzy Agent

The fuzzy task agents are able to learn the user behavior and model it by monitoring the user actions. The FTA then creates fuzzy-based linguistic models which could be evolved and adapted online in a life-learning mode. This fuzzy-based system could be used to control the environment on the user behalf and to his satisfaction. The intelligent approaches used within the agents should have low computational overheads to effectively operate on the embedded hardware platforms present in the everyday environments (such as fridges, washing machines, and mobile phones) which have small memory and processor capabilities. In addition, the intelligent approaches should allow for real-time data mining of the user data and create on-the-fly updateable models of the user preferences that could be executed over the pervasive network. Moreover, there is a need to provide an adaptive lifelong learning mechanism that will allow the system to adapt to the changing environmental and user preferences over short- and long-term intervals. In all cases it is important that these intelligent approaches represent their learnt decisions and generate the system's own rules in a form that can be easily interpreted and analyzed by the end users (Hagras et al., 2007). There is a need also to provide robust mechanisms that will allow handling the various forms of uncertainties so that the system will be able to operate under the varying and unpredictable conditions associated with the dynamic environment and user preferences.

Inhabited AmI spaces face huge amount of uncertainties which can be categorized into environmental uncertainties and users' uncertainties. The environmental uncertainties can be due to the following:

- the change of environmental factors (such as the external light level, temperature, time of day) over a long period of time due to seasonal variations;
- the environmental noise that can affect the sensors measurements and the actuators outputs;
- wear and tear which can change sensor and actuator characteristics.

The user uncertainties can be classified as follows:

- intra-user uncertainties that are exhibited when a user decision for the same problem varies over time and according to the user location and activity. This variability is due to the fact that the human behavior and preferences are dynamic and they depend on the user context, mood, and activity as well as the weather conditions and time of year. For the same user, the same words can mean different things on different occasions. For instance the values associated with a term such as "warm" in reference to temperature can vary as follows: depending on the season (for example, from winter to summer), depending on the user activity within a certain room and depending on the room within the user home and many other factors;
- inter-user uncertainties which are exhibited when a group of users occupying the same space differ in their decisions in a particular situation.

This is because users have different needs and experiences based on elements such as age, sex, and profession. For instance the users might disagree on aspects such as how warm a room should be on any given day.

Thus it is crucial to employ adequate methods to handle the above uncertainties to produce models of the users' particular behaviors that are transparent and that can be adapted over long time duration and thus enabling the control of the users' environments on their behalf.

Fuzzy logic systems (FLSs) are credited with being adequate methodologies for designing robust systems that are able to deliver a satisfactory performance when contending with the uncertainty, noise, and imprecision attributed to real-world settings (Doctor et al., 2005). In addition, an FLS provides a method to construct controller algorithms in a user-friendly way closer to human thinking and perception by using linguistic labels and linguistically interpretable rules. Thus FLSs can satisfy one of the important requirements in AmI systems by generating transparent models that can be easily interpreted and analyzed by the end users. Moreover, FLSs provide flexible representations which can be easily adapted due to the ability of fuzzy rules to approximate independent local models for mapping a set of inputs to a set of outputs. As a result, FLSs have been used in AmI spaces as in Doctor et al. (2005), Rutishauser et al. (2005), Hagras et al. (2007).

Recently, type-2 FLSs, with the ability to model second-order uncertainties, have shown a good capability of managing high levels of uncertainty. Type-2 FLSs have consistently provided an enhanced performance compared to their type-1 counterparts in real-world applications (Coupland et al., 2006; Hagras et al., 2007). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] (Mendel, 2001). There are two variants of type-2 fuzzy sets – interval-valued fuzzy sets (IVFS) and generalized type-2 fuzzy sets (GFS). In an IVFS the membership grades are across an interval in [0,1] and have the third dimension value equal to unity. In the case of a GFS the membership grade of the third dimension can be any function in [0,1]. Most applications to date use IVFS due to its simplicity; however, recent work has allowed GFS to be deployed efficiently.

It has been shown that IVFS-based type-2 FLSs can handle the environmental uncertainties and the uncertainties associated with a single user in a single room environment and that type-2 FLSs can outperform their type-1 counterparts (Hagras et al., 2007). However, no work has tried to approach the challenging area of developing AmI spaces that can handle the environmental uncertainties as well as the intra- and inter-user uncertainties in an environment that has multiple rooms populated by multiple users.

There are many frameworks for dealing with uncertainty in decision-making including, primarily, those based on probability and probabilistic (Bayesian) reasoning. As an aside, we emphasize that we do not claim that fuzzy-based methods are any better or any more effective than any other uncertainty handling frameworks, rather we claim that some methods are more appropriate in certain contexts. In our experience, fuzzy methods have proven to be more effective than other methods when applied in AmI spaces. This is because the fuzzy methods provide a framework using linguistic labels and linguistically interpretable rules which is very important when dealing with human users.

We shall employ the use of type-2 fuzzy logic to model the uncertainties in such AmI spaces. Consider an example of a central heating system for a living space occupied by two users. In such a situation each user's concept of cold has to be modeled throughout the year. There will be seasonal variations affecting each user's idea of what is cold, an example of intra-user variation. Each individual user will have his notion of what temperatures constitute cold, an example of inter-user uncertainty. Modeling either of these uncertainties can be accomplished using a number of existing techniques. The novel challenge with this is to model and cope with the uncertainties created by the dynamic relationship between the interactions of multiple users, each with individual preferences that change over time. For example, Figure 3(a) and (b) shows the use of type-1 fuzzy systems to depict the differences between two users (p1 and p2) concept of cold for the spring/summer and autumn/winter periods. Figure 3(c) and (d) shows how interval type-2 fuzzy sets might model each user's concept of cold throughout the year. Figure 3(e) shows how a general type-2 fuzzy set might encompass both the inter- and intra-user uncertainties about what cold is by employing the third dimension, where the different gray levels correspond to different membership levels in the third dimension.

The embedded agents learn and adapt to the user behaviors in AmI spaces using our type-2 incremental adaptive online fuzzy inference system (IAOFIS) technique. IAOFIS is an unsupervised data-driven one-pass approach for extracting type-2 fuzzy MFs and rules from data to learn an interval type-2 FLC that will model the user's behaviors.

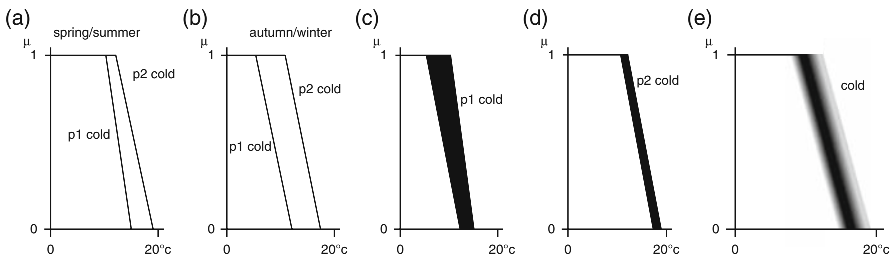

Figure 3. Fuzzy sets modeling the linguistic label cold.

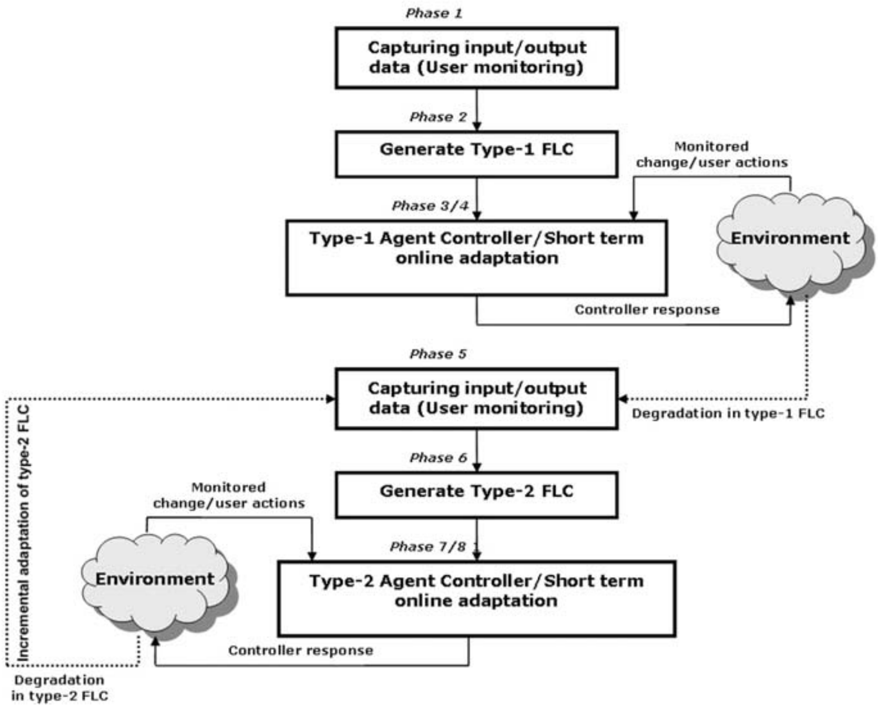

Figure 4. Flow diagram showing the main phases of IAOFIS.

The IAOFIS approach consists of eight phases of operation as described in Figure 4. In Phase 1, the system monitors the user interactions in the environment for a specific time (3 days in case of our experiments) to capture input/output data associated with the user actions. In Phase 2 the system learns from the data captured in phase 1 the type-1 MFs and rules needed to form a type-1 FLC that can effectively model the user's behaviors under the specific environmental conditions during phase 1. The approach used for learning this type-1 FLC can be found in Doctor et al. (2005), where the method for learning the type-1 MFs is based on a double clustering approach combining fuzzy-C-means (FCM) and agglomerative hierarchical clustering. In Phase 3, the learnt type-1 FLC operates in the environment to satisfy the user preferences under the faced environmental conditions. The type-1 FLC can handle the short-term uncertainties arising from slight sensor noise or imprecision as well as slight changes in environmental conditions such as small changes in temperature and light level due to variations in the weather conditions. The type-1 agent can also adapt in the short term as shown in Phase 4 by updating the FLC rule base through adding or adapting rules to reflect the user preferences associated with the encountered environment conditions. However, over a long period of time, the long-term uncertainties caused by seasonal changes and the associated changes in user activity will result in a significant deviation of the type-1 MFs parameters (associated with the linguistic labels for the input and output variables) from those initially modeled by the type-1 FLC. So whatever adaptations occur to the rules, this will not improve the system performance as the MFs values attached to the linguistic labels (which are the antecedents and the consequents of the rules) no longer reflect the current environment and user preference. This will cause the performance of the type-1 FLC to degrade which can be gauged by the increase in user interaction with the system to override the type-1 FLC outputs to try to adapt the system to his desires; this reflects the user's dissatisfaction. When the type-1 FLC sufficiently degrades a system adaptation trigger is activated and the system goes to Phase 5 in which the user is re-monitored again under the new environmental conditions for a specific time interval (again 3 days in case of our experiments). The system then goes to Phase 6 in which the agent learns the interval type-2 MFs and rules to form an interval type-2 FLC. The system then moves to Phase 7 in which the type-2 FLC controls the user environment based on the user-learnt behaviors and preferences. The type-2 agent can adapt in the short term as shown in Phase 8 by updating the type-2 FLC rule base through adding or adapting rules to reflect the user preferences associated with the encountered environment conditions. However after an extended period of time, new uncertainties arise due to the continual seasonal changes which occur in the environment, hence the type-2 MFs parameters associated with the linguistic labels change which will cause the performance of the type-2 FLC to degrade. The agent again enters a monitoring mode in Phase 5 to re-monitor the user behavior under new environmental conditions, the agent will then incrementally adapt the type-2 FLC by generating a new set of type-2 MFs and rules to take into account the current and previous uncertainties. Our system can therefore incrementally adapt the type-2 FLC in a lifelong-learning mode so that its type-2 MFs and FOUs capture all the faced uncertainties in the environment during the online operation of the agent. The adapted type-2 FLC also retains all the previously learnt user behaviors captured in the FLC rule base.

6. Adaptive User Interaction

An important dimension of adaptation is the interaction between the AmI space and the user. From a user's point of view, computer-based systems nowadays still are somehow notional and incomprehensible. In order to bridge this gap, it is necessary to establish a natural and in various ways adaptive interface between users and ambient ecologies. Displays and screens are ideally suited to give an overview about different kinds of structured information. For some other tasks we assume spoken interaction within the ambient ecology as first choice. To bring the advantages of the various ways of interaction together we consider certain rules for choosing the best input and output modality as provided:

- Output modality:
  - depends on privacy rules, e.g., somebody is entering the room;
  - depends on context, e.g., what devices are available;
  - depends on input mode;
  - depends on the information itself, e.g., music needs audio output.
- Input modality:
  - user explicitly chooses input modality;
  - depends on available resources;
  - depends on privacy rules, e.g., in public the user may not want to use speech input.

We argue that it is not a trivial problem to automatically choose the most suitable modality for interaction, whereas this choice is also an important part of adaptation. Since interaction is a huge area of research the remainder of this section exemplifies adaptive interaction by scoping on spoken interaction.

A spoken dialogue system within ambient environments may provide three classes of spoken interaction (Figure 5):

1 A main dialogue giving users the ability to control devices and services within the ambient ecology by the use of standard commands.

2 Pro-active dialogues, which are initialized by the ambient ecology.

3 On-the-fly special purpose dialogues, which are generated depending on the context and give the user a change to negotiate with the system and to ask or to submit further information.

The characteristics of the main dialogue mainly depend on the devices and services registered to the environment. Some commands to control the environment are as follows:

- control lights (e.g., “Turn on the lights!”);
- control temperature (e.g., “I feel cold.”);
- control blinds (e.g., “Black out the room!”);
- locate things (e.g., “Where are my keys?”).

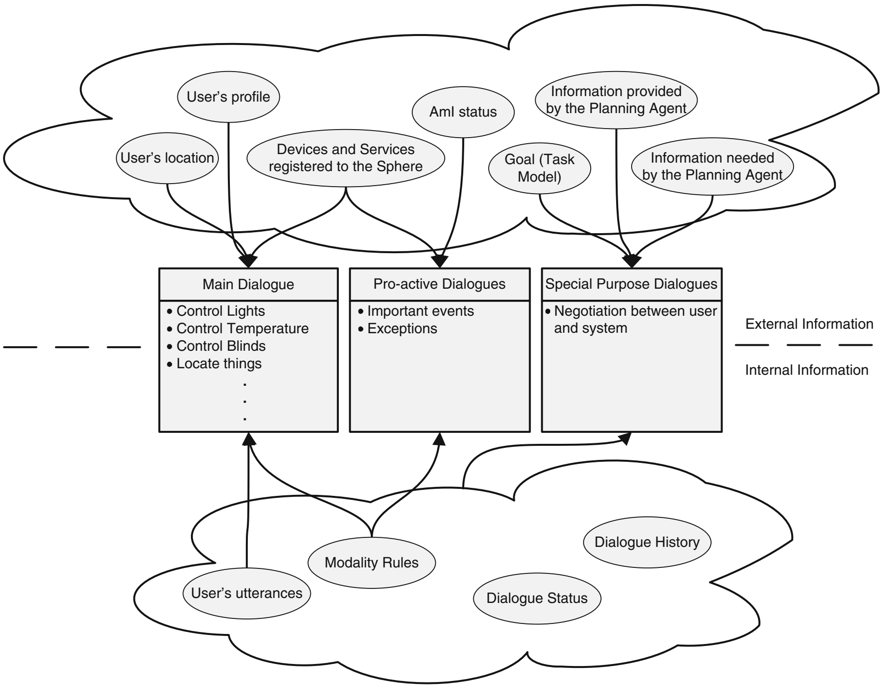

Figure 5. The main values needed for interaction and the corresponding dialogue classes.

If there are, for example, no lights available, the main dialogue will provide no commands for controlling any lighting. Thus the dialogue adapts to the infrastructure. If there are lights available an example grammar in Nuance GSL for controlling lighting may look like the following:

```
[COMMON_GRAMMAR:str]{<choice $str>}
COMMON_GRAMMAR[
[(?([turn switch] on the)lights) (lights on)] {return("lights-on")}
]
```

To increase the performance of the recognition it is possible to simplify the grammar by the use of formerly learnt “favorite commands” which may be stored in the user profile. Thus the dialogue adapts to the user. Another way of adapting may be utilized by tracking the user's location. The actual location is crucial to the main dialogue since the user's topics are in many cases directly subject to the physical surroundings. We plan to analyze if there are any other context information, e.g., time or ambience within the environment, needed for adaptive dialogue management. Apart from the mentioned external information, the main dialogue also depends on internal information such as the user's utterances and the modality rules as aforementioned.

The main dialogue could also be used for the semi-tacit configuration of the sphere mentioned in Section 2. It is imaginable to provide a main dialogue running in configuration mode by following the same ideas and using the same information resources as in standard mode.

Another class of spoken interaction are pro-active dialogues. To interact pro-actively, the ambient ecology has to decide what information is important enough to justify a pro-active system activity. The main external information underlying such a decision could be deduced from the system status and from a device or service itself. We differentiate between

- important events, e.g., AmI wakes up the user earlier because of a traffic jam to prevent him of being late at work;

- exceptions, e.g., AmI should be able to inform the user about the occurrence of exceptions which may cause damage to the environment or the residents.

The only internal information required for generating those pro-active dialogues is related to the modality rules.

Most difficult to handle is the class of special purpose dialogues. Dialogues belonging to this class give the user a change to negotiate with the system and to ask or to submit further information. The special purpose dialogues are generated depending on the external information provided by the sphere's task model itself and by the planning agent. To process a plan the planning agent on the one hand provides information which has to be communicated to the user and on the other hand requests information to accelerate planning or even to proceed with the plan. A short example scenario explains this in more detail:

Find a recipe depending on Suki's preferences and on available ingredients - Suki is hungry and wants the AmI to help him finding a recipe for a meal he would like to eat. Suki does not have an overview of the available ingredients but the Ambient Ecology does. The PA starts to find out which recipes are possible to cook with the available ingredients. It requires more information about what Suki would like and therefore a dialogue with Suki starts.

SDS: Do you rather want a warm dinner or just a sandwich?

Suki: I'm really hungry - I would like to have dinner!

To enable AmI to interact in this manner we need a more sophisticated dialogue model. In this case the internal information consists of the user's utterance, modality rules, a dialogue history, and a dialogue status. This dialogue status can, for example, follow the ideas mentioned in Larsson and Traum (2000).

To stay compatible with different well-established speech dialogue systems (TellMe, Voxeo, Nuance) we can generate dialogues using VoiceXML (Oshry et al., 2000) since this is the most common description language for speech dialogues. To minimize the limitations of VoiceXML we can use concepts similar to the popular AJAX web technologies to establish dynamic dialogues.

7. Conclusion

In this chapter we presented an architecture that can support adaptation of an intelligent environment to the requirements of specific user tasks. We model user tasks that serve a specific goal as an activity sphere and we consider that the system supporting the sphere must execute its tasks by using the services provided by the devices of an ambient ecology – we use this term to describe devices that can communicate and collaborate on demand.

The proposed architecture uses an ontology as the centralized repository of knowledge and information about the ambient ecology and a set of intelligent agents, which access and manipulate the ontology. The ontology is formed by aligning the local ontologies (or meta-data) of the devices with the concrete task descriptions, the user profile, and any policy ontologies possessed by the agents or the environment. The ontology is managed by the ontology manager and the whole activity supporting system is managed by the sphere manager.

Three kinds of adaptation can be achieved:

- task adaptation, whereby the fuzzy task agent monitors the way the user interacts with the devices of the ambient ecology and adapts its rules (and the ontology) accordingly;
- plan adaptation, in the case that the configuration of the ambient ecology changes, whereby the planning agent re-computes the concrete task descriptions based on the new configuration – plan adaptation requires the re-alignment of the sphere ontology;
- interaction adaptation, whereby the interaction agent, based on the ontology, calculates on a case basis the best way to interact with the user.

Ontology-based pervasive systems have already been presented in the literature. Among these

- the CADO (context-aware applications with distributed ontologies) framework (De Paoli and Loregian, 2006) relies on distributed ontologies that are shared and managed in a peer-to-peer fashion, so as to ensure semantic interoperability via the process of ontology merging;
- CoBrA (context broker architecture) (Chen et al., 2003) uses a collection of ontologies, called COBRA-ONT, for modeling the context in an intelligent meeting room environment. These ontologies expressed in the Web Ontology Language (OWL) define typical concepts associated

with places, agents, and events and are mapped to the emerging consensus ontologies that are relevant to the development of smart spaces;

- GAIA (Roman et al., 2002) uses ontologies as an efficient way to manage the diversity and complexity of describing resources, that is, devices and services. Therefore, these ontologies are beneficial for semantic discovery, matchmaking, interoperability between entities, and interaction between human users and computers.

All these ontology-based systems use static heavyweight domain ontologies to support ubiquitous computing applications. These ontologies are used to represent, manipulate, program, and reason with context data and they aim to solve particular ubiquitous computing problems, such as policy management, context representation, service modeling and composition, people description, and location modeling. However, in the ubiquitous computing domain, it is difficult for applications to share changing context information, as they will have to constantly adapt to the changes.

Our approach is different because it is based on (a) the existence of heterogeneous smart components (devices, services) within an ambient intelligence environment, (b) the fact that these components maintain and make available local representations of their self and state, and (c) their ability to communicate and collaborate. Thus, we propose a bottom-up scheme which maintains the independence of task description from the capabilities of the ambient ecology at hand and at the same time achieves adaptation at the collective level, without the need to manipulate local device ontologies.

References

Chen, H., Finin, T., and Joshi, A. (2003). An Ontology for Context-Aware Pervasive Computing Environments. *Knowledge Engineering, Special Issue on Ontologies for Distributed Systems*, 197–207.

Coupland, S., Gongora, M., John, R., and Wills, K. (2006). A Comparative Study of Fuzzy Logic Controllers for Autonomous Robots. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006)*, pages 1332–1339, Paris, France.

De Paoli, F. and Loregian, M. (2006). Context-Aware Applications with Distributed Ontologies. In *Proceedings of the 18th International Conference on Advanced Information Systems Engineering*, 869–883.

Doctor, F., Hagras, H., and Callaghan, V. (2005). An Intelligent Fuzzy Agent Approach for Realising Ambient Intelligence in Intelligent Inhabited Environments. *IEEE Transactions on System, Man, and Cybernetics, Part A*, 35(1):55–65.

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., and Burgelman, J.-C. (2001). Scenarios for Ambient Intelligence in 2010. Technical Report, IST Advisory Group Final Report, European Commission.

Euzenat, J. and Schvaiko, P. (2007). *Ontology Matching*. New York: Springer.

Goumopoulos, C. and Kameas, A. (2008). Ambient Ecologies in Smart Homes. *The Computer Journal*, advanced access published online at doi: 10.1093/comjnl/bxn042.

Gruber, T. (1993). A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199–220.

Hagras, H., Doctor, F., Lopez, A., and Callaghan, V. (2007). An Incremental Adaptive Life Long Learning Approach for Type-2 Fuzzy Embedded Agents in Ambient Intelligent Environments. *IEEE Transactions on Fuzzy Systems*, 15(1):41–55.

Kameas, A., Bellis, S., Mavrommati, I., Delaney, K., Colley, M., and Pounds-Cornish, A. (2003). An Architecture that Treats Everyday Objects as Communicating Tangible Components. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 115–122. IEEE CS Press.

Kameas, A., Mavrommati, I., and Markopoulos, P. (2005). Computing in Tangible: Using Artifacts as Components of Ambient Intelligence Environments. In Riva, G., Vatalaro, F., Davide, F., and Alcañiz, M., editors, *Ambient Intelligence*, pages 121–142. IOS Press.

Larsson, S. and Traum, D. (2000). Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 6:323–340.

Mendel, J. (2001). *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River NJ: Prentice-Hall.

Oshry, M., Auburn, R., Baggia, P., Bodell, M., Burke, D., Burnett, D., Candell, E., Carter, J., McGlashan, S., Lee, A., Porter, B., and Rehor, K. (2000). Voice Extensible Markup Language (VoiceXML) Version 2.1. *W3C – Voice Browser Working Group*.

Remagnino, P. and Foresti, G. L. (2005). Ambient Intelligence: A New Multidisciplinary Paradigm. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(1):1–6.

Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., and Nahrstedt, K. (2002). Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing*, 1(4):74–83.

Rutishauser, U., Joller, J., and Douglas, R. (2005). Control and Learning of Ambience by an Intelligent Building. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(1):121–132.

Seremeti, L. and Kameas, A. (2008). Ontology-Based High Level Task Composition in Ubiquitous Computing Applications. In *Proceedings of The*

4th International Conference on Intelligent Environments (IE08), Seattle, USA, 1–5.

Zaharakis, I. D. and Kameas, A. (2008). Engineering Emergent Ecologies of Interacting Artifacts. In Lumsden, J., editor, *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*. IGI Global, 364–384.