Symbiotic Ecologies in Next Generation Ambient Intelligent Environments

H. HAGRAS[1],C. WAGNER[2], A. KAMEAS[3], C. GOUMOPOULOS[3],A. MELIONES[4],L. SEREMETI[3],Y. BELLIK[5],G. PRUVOST[5], T. HEINROTH[6] AND W. MINKER[6]

This paper describes a novel approach to realize symbiotic ecologies within Next Generation Ambient Intelligent Environments (NGAIEs). The proposed approach comprises novel ontology and agent technologies allowing for adaptation on a variety of levels. The metaphor of symbiotic ecologies reflects a meaningful integration of relevant entities (i.e., services, devices, agents) and information within NGAIEs to accomplish a specific user's task by relying on the symbiotic relationship of the user and his/her intelligent environment. We adopt a service-oriented architecture, combined with (a) intelligent agents that support adaptive task realization and enhanced human-machine interaction and (b) ontologies that provide knowledge representation, management of heterogeneity at user and device level, semantically rich resource discovery and adaptation using ontology alignment mechanisms. In this article, we analyse heterogeneity concerning user behaviour and adaptive user interaction modelling. Furthermore, we focus on heterogeneity regarding the representation of the states of entities and their availability over different networks. The paper will also report on the deployment of a system prototype in a real world setting which is the intelligent flat (iSpace) at the University of Essex.

Keywords: Fuzzy technology, Ontologies, Agent-based SOA

---

1. INTRODUCTION

After more than ten years of research, it has now become clear that *Ambient Intelligence (AmI)* is not yet another technological development, but a new paradigm that will potentially affect all domains of our society. Computation and networking have already been embedded into devices and everyday environments, supporting the collection, processing and exchange of information, thus providing situation and activity recognition, context representation, proactive behaviour, collaboration and resource management in an increasingly intelligent way. Our environments are being transformed into *Ambient Intelligent Environments* (AIEs) populated with smart communicating objects, which are able to perceive the environment, act upon it, process and store data, manage their local state, communicate and exchange data. In addition, the AIEs provide communication infrastructure and context aware services that are used by ubiquitous computing systems and applications.

AIE architectures describe systems that consist of a small number of networked objects or network services and systems that are aimed at providing transparent context dependent access to these resources to higher level ubiquitous computing applications. The next step is the design and development of totally adaptive ubiquitous computing systems which are able to consistently operate in heterogeneous pervasive AIEs. Adaptive software systems use mechanisms in order to define and maintain their identity, regulate their operation and change their structure as they establish different types of coupling with the environment. *In this Next Generation of AIEs* (NGAIEs) [Abowd, G. et al.], objects will collaborate in the realization of ubiquitous computing applications within the hosting AIE where the structure of such applications may dynamically change and their efficient operation will depend on the orchestration of heterogeneous services. In addition, the system will inherently exhibit increasingly intelligent behaviour, provide optimized

---

Authors' address: [1]The Computational Intelligence Centre, University of Essex, United Kingdom,[2]Horizon Digital Economy Research Institute, School of Computer Science, Univ. of Nottingham, UK,[3] The Hellenic Open University, Greece, and DAISY research unit, CTI, Greece,[4]Dept. of Digital Systems, University of Piraeus, and inAccess Networks S.A., Greece,[5]LMSI/CNRS, Paris-South University, France,[6]University of Ulm, Germany

resource usage and support consistent functionality together with human-centric operation (humans, as opposed to mere users, have increased requirements from a system including intuitive interaction, protection of privacy, fault-tolerance, etc.). Consequently, the realization of NGAIEs brings up various research challenges, such as [DUCATEL, K. et al.]:

—*Heterogeneity*: In the general case, artefacts will be heterogeneous, as they will originate from different manufacturers, and they will probably use custom communication protocols and proprietary information representation schemes. Some of them may easily interoperate with existing networks whereas some may not even integrate at all. Therefore, NGAIEs should provide the means to facilitate inter-operability, for example via defining an abstract interface or an application that can access other devices and using their functionalities [DOCTOR, F. et al.].

—*Transparency*: Users shouldn't be aware of the composition or internal state of AIE components in order not to become overwhelmed by the AIE individual settings, the technological enhancements or the infrastructural changes according to the AmI paradigm. They should rather be supported in developing trust in technology as an enabler of their everyday activities that makes their life easier. Instead of detailed description of state and constant interaction to reassure state changes, NGAIEs should provide an adaptive degree of pro-activity combined with self-explanation and other self-* properties, so as to transparently and optimally (with respect to available resources) support people's tasks [HAGRAS, H. et al.].

—*Discovery and Management*: NGAIEs will consist of hundreds to thousands of heterogeneous artefacts interacting via various networks. In fact, their structure will resemble that of a complex system, thus a combination of global resource discovery and local resource management will be required. It is worthwhile to note that interaction management at a global scale will not be an easy target to achieve; yet, hierarchical management schemes are promising candidates and they have already been adopted by various communication protocols [HAGRAS, H. et al.].

—*Intelligence*: Intelligence, as the primary means to achieve adaptation, will appear at various levels. For example, local resource management may require decision making mechanisms and even embedded intelligent agents. At a system scale, multi-agent systems, which use semantically rich descriptions, learning mechanisms and possibly cognitive functions (such as perception, homeostasis, etc.), will be embedded in NGAIEs. These agents are able to recognize the users and can autonomously program themselves to meet the user's needs and preferences by learning from their behaviour to control the environment on their behalf, thus reducing the cognitive load associated with the user needing to configure and program ubiquitous computing applications [HAGRAS, H. et al., HAGRAS, H. et al. 2007].

In this paper, we shall present the ATRACO (Adaptive and TRusted Ambient eCOlogies) approach towards the realization of NGAIEs. To start with, the definition of *Ambient Ecologies* (AE) conceptualizes the spaces populated by interrelated and therefore connected devices and services, various environments and people [GOUMOPOULOS, C. et al.]. In our approach, we establish the term *Symbiotic Ecologies* (SE) to describe what happens when human and AE collaborate and benefit from each other. A generalization of SE is what we call Next Generation Ambient Intelligent Environments (NGAIEs). NGAIEs support the realization of *Activity Spheres* that are modelled as dynamic and purposeful projections of SE.

In contrast to previous research, within the ATRACO approach [ATRACO , HEINROTH, T. et al.] we adopt a unique standpoint in modelling and realizing NGAIEs. We assume that various NGAIEs are already available; each of them hosting a dynamically changing set of heterogeneous smart objects (e.g., interactive tables, networked cups and lamps, RFID enabled closets and refrigerators, adaptive TVs and air conditioning devices, etc.), and components (e.g., temperature, humidity, brightness, motion, etc. sensors). They, nevertheless, contain heterogeneous descriptions of their capabilities and services that can only be accessed from but cannot be modified by other components. Thus, these objects can collaborate in the realization of ubiquitous computing applications within the hosting NGAIE, but the structure of these applications may dynamically change and their efficient operation depends on the orchestration of heterogeneous services. In order to achieve task-based collaboration amongst them, 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 NGAIE.

Our modelling approach is described in Section 2. Section 3 gives an overview of the ATARCO architecture and its basic principles and outlines the basic components whose interactions provide the adaptive behaviour of ATRACO system. ATRACO supports several dimensions of adaptation within the SE, namely *Network Adaptation* (Section 4), *Structural Adaptation* (Section 5), *User Behaviour Adaptation* (Section 6), *Semantic Adaptation* (Section 7), and *User Interaction Adaptation* (Section 8). In order to achieve these levels of adaptation, we use a set of intelligent agents to support adaptive workflow management, task realization and enhanced human-machine interaction, based on a dynamically composed ontology of the properties, services and state of the NGAIE resources. In this paper, we shall discuss how the proposed approach deals with heterogeneity and supports a unique multi-dimensional adaptation capability in NGAIEs. Then, we shall briefly describe how the proposed approach was realised in a prototype (Section 9) which was deployed, validated and evaluated in real NGAIEs (Section 10). Finally, we provide our conclusions and the lessons learnt (Section 11).

2. THE ATRACO MODELLING APPROACH

As we move from physical to digitized spaces, some of the existing real world components, concepts or metaphors will have to be adapted accordingly. We use the Ambient Ecology (AE) term to denote the set of heterogeneous artefacts with different capabilities and services that reside within the NGAIE [GOUMOPOULOS, C. et al.]. AEs support people in achieving their goals and carrying out new tasks as well as old tasks in new and better ways. People will realize tasks using the resources offered by the services and devices of the AE. A Symbiotic Ecology (SE) is a generalization of an AE, in a sense that (a) it contains mechanisms that cause the emergence of systemic features (such as identity, preservation, adaptation, etc.) and (b) it develops coupling mechanisms (realized through interactions) with another SE or, in general, with the environment, aiming at adaptation, optimization and evolution. An SE can also be considered as a "society" of interacting individuals (including objects and services but also users, agents, and robots).

The notion of “Bubble” has been used to describe a temporarily defined space that can be used to limit the information coming into and leaving the digital domain [BESLAY, L. et al.], which itself constitutes a “digitization” of the definition of personal space described as a “soap bubble” [SOMMER, R. et al.]. A bubble is supposed to gather all the interfaces, formats and agreements etc. together which are needed for the management of personal, group and public data coupled with informational interactions. In other words, it may be described as a semi-transparent membrane that can be tuned to function differently depending on the direction of the movement of data.

Based on the aforementioned idea, we have defined the concept of an *Activity Sphere* (AS), to be both the model and the realization of the set of information, knowledge, services and other resources required to achieve an individual goal within an NGAIE. By the inspiration of object-oriented approaches, an AS expands the bubble notion to contain not only the data and models, but also the associated processes and other resources that create, use or otherwise affect the data, leading to the specification of autonomous and coherent entities, which can adaptively execute on a changing infrastructure. Initially, an AS is an abstract model of a plan to achieve some goals, expressed in particular tasks or in workflow description language. When the AS is identified within the context of an SE, then the abstract plan becomes concrete, that is, abstract descriptions of tasks (i.e. set temperature to high, set music to ambient) become concrete task descriptions (i.e. set the temperature of room air-condition to 25 C, use the CD player to play a record by Yanni), which can be realized by specific service invocations (bindings) from specific devices (i.e. a call to the temperature setting service offered by the air-conditioner with parameter "25 C", a call to the disk play service offered by the CD player with parameter "Yanni"). An AS contains a set of interrelated tasks of such kind thereby leading to the achievement of non-trivial user goals. It should be noted that in case of moving to a different room, the abstract task remains unchanged, but the concrete task and the binding change depending on the services offered by the SE in the new room. The concretization of task descriptions and the realization of bindings are carried over by the ATRACO system (i.e. an attempt to locate the air-conditioner and the CD player in the new room will be done by the ATRACO system).

On the whole, an AS has a dual hypostasis: first, it is a semantically rich model of an entity's specific goal and its associated tasks together with a specification of the realization of this model; second, it is the orchestration of specific services, data and knowledge that realize this specification within the context of a particular SE. The abstract task model can be stored in the AS owner profile, in a server, in some web location, etc., which means that it is portable. As the task model is eventually realized by orchestrating the specific services offered by SE components, one can say that any SE is used as the platform, which is the virtual machine that can "realize" an AS. By including mechanisms that maintain its internal structure and at the same time support adaptation, the sphere can deal to a certain extent with internal and external changes by adapting its structure and functionality. As mentioned above, the AS can be realized, in an adaptive way, within different SE, each time exploiting the available resources. This is true in both cases where the AS is moved to a new SE, or where the composition of the current SE changes dynamically due to some components ceasing to be available, or new components being added. Depending on the severity of the changes in the SE, the realization of the AS within the new SE may or may not require re-planning.

The mechanism we propose to achieve the different kinds of adaptation implied by this approach focuses on the management of knowledge that is encoded in multi-layered ontologies, which are used by intelligent agents. For example, in order to set the temperature of the air-conditioner, the ATRACO system will first access the user profile ontology and determine the value range for the label "high"; then, it will access the device ontology, where the services and the current state of the air-conditioner are stored, locate the temperature entity and monitor its value. An agent will make sure that (a) the value is kept within the acceptable range, as determined by the meaning of "high" and (b) if this meaning changes, then the new range will be recorded in the user profile ontology and the concrete plan will be adapted accordingly. Hence, it can be drawn that knowledge bridges the gaps between the levels of a layered architecture forming an AS. The levels of knowledge we have defined in order to address the layered architecture are as follows:

—HCI: represents the knowledge necessary to carry out intuitive dialogue between an AS and its human owner;

—*Semantic*: represents the interpretation of AS related knowledge within a specific context;

—*Behaviour*: encodes the dynamic operation of the AS (that is, the properties, capabilities and services offered by its components);

—*Structure*: encodes the architecture (that is, the components and the relationships between them) of the AS, or, recursively, of any of its components;

—*Communication*: is the low level knowledge required to exchange messages between the AS components; such messages are deprived of context or interpretation (but may constitute a description of context that is to be exchanged between devices).

These levels also describe the facets of concreteness and abstraction towards the user with respect to his/her perception of the system. On the top level, the system is concrete from a human point of view and abstract from a system point of view. As the user shifts focus towards the bottom levels, the system components become concrete, but their operation becomes vague. Despite the distribution of computing components and services within NGAIEs, the user's interaction with a ubiquitous computing system will not cease to be task-centric: in order to accomplish an activity, users are still required to carry out specific tasks, using the skills, tools and information available in their heads, or in the environment.

In order to achieve dynamic task-based coupling between AS and NGAIE, one has to deal with heterogeneity in description of resources, while achieving independence between a task description and its respective realization within a specific NGAIE. Ontologies are used to tackle the semantic heterogeneity that arises in NGAIEs and to provide a common repository of system knowledge, policies and states. Heterogeneity is an intrinsic NGAIE feature that causes problems in practical AS operation. For each level, we deal with a specific type of heterogeneity. For instance, HCI realized by a spoken dialogue system is confronted with the heterogeneity of human speech. How to handle this heterogeneity is still an unresolved research issue. On the other hand, ontology alignment provides a solution for representing heterogeneous knowledge on the Semantic level. Likewise, advanced Fuzzy technologies can be used to adapt the behaviour of the AS to the heterogeneous behaviour of the user. On the structural level, ontologies can be used to provide knowledge about the changing population of entities within the AS and finally on the communication level, adaptive middleware and system hardware can be used to maintain the technical interrelation of AS entities.

3. THE ATRACO ARCHITECTURE

ATRACO supports the deployment and execution of applications that need to be adapted and reconfigured in dynamic environments. The need for adaptation and reconfiguration calls for a modular design approach, which the *Service-Oriented Architecture* (SOA) paradigm tends to provide [ERL, T. 2005 et al.]. Following this architectural style, each device provides services through which other components can obtain information or control its behaviour. When an application has to be adapted, either during application transition to a new environment or when a device running a service fails, a description of the structure of the application, which is modelled as a workflow of abstract services, is used by an adaptation module which makes use of ontologies, context information and defined policies to generate a new structure for the adapted application. We adopt SOA both at the resource level (to support resources, such as devices, sensors and context to become integrated in applications) and the system level (to combine system services in the ATRACO system in order to support ubiquitous computing applications).

Contemporary software techniques complying with the SOA architectural paradigm, such as OSGi, UPnP, and the Web Services do not meet the adaptability and interoperability challenges of NGAIEs on their own. In the first case, SOA provides little support on how adaptive services can be used to allow people to transparently interact with an AmI environment. The challenge is to automate the service composition process, so that the services offered to the users adapt dynamically to the task they wish to perform and its context. In the second case, current solutions provide little support for semantic-based interoperability, only dealing with interaction between services based on syntactic description for which common understanding is hardly achievable in an open environment.

In ATRACO we propose a combination of the SOA model with Agents and Ontologies (see Figure 1). The ATRACO architecture consists of active entities (agents and managers), services, passive entities and ontologies.

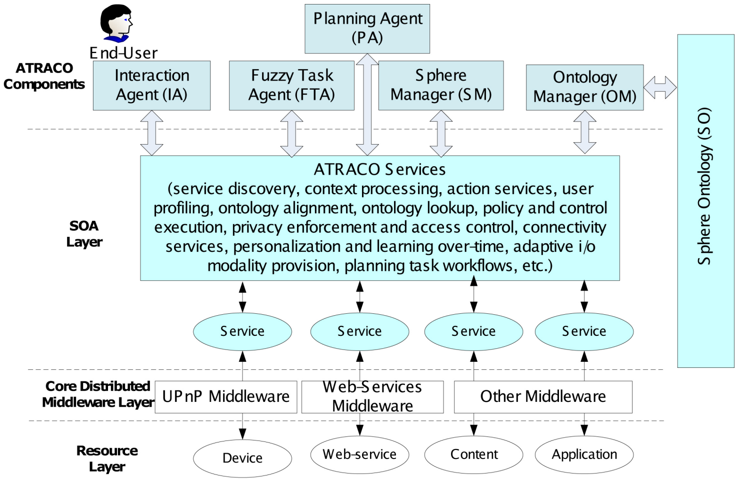


Figure 1: Active bundle structure.

3.1 Active Entities

Active entities are managers and agents. The managers (*Sphere Manager* (SM) and *Ontology Manager* (OM)) are responsible for the formation of the AS and for keeping the Sphere Ontology up-to-date. The agents (*Planning Agent* (PA), *Fuzzy logic based Task Agent* (FTA) and *Interaction Agent (IA)*) are responsible for task planning, automated adaptation, resolving conflicts, interacting with the user, and in general supporting the users achieving their goals. In this paper, we elaborate only on two of the agents i.e. IA and FTA. In another work [BIDOT, J. et al.], we have presented the functionality of the PA, which -at design time- is responsible for (i) creating a planning problem that consists of the user's goal to be achieved and the services currently offered by the intelligent environment, (ii) solving the planning problem using Hierarchical Task Network and Partial-Order Causal-Link planning techniques, and (iii) generating abstract service workflows from the planning decisions taken to find solution plans. The rest of the system described in this paper supports adaptation and application structure reconfiguration at runtime.

3.2 Passive Entities

Passive entities are devices such as interaction devices (touch screens, speakers, microphones, etc.), actuators and sensors (including televisions, radio receivers and HVACs, etc.), services such as remote or external web-services (i.e., online banking) and local or internal services (i.e., personal calendar). They are usually triggered by agents and therefore behave passively.

3.3 Ontologies

Ontologies are used to provide semantic modelling by expressing the basic terms and their relations in a domain, task or service. Thus, they constitute an extensible and flexible way of tackling the semantic heterogeneity that arises in NGAIEs by providing agents a common repository of system knowledge, policies and state. Moreover, through the semantics they convey, ontologies are used in order to address two important goals of NGAIE design, namely the increase of the amount of knowledge available to the system and the minimization of the inaccuracy of knowledge as well as the ambiguity regarding the interpretation of the shared information. Hence, despite their heterogeneous representations of the world, NGAIE components are able to interact successfully through a common communication channel.

There are two main kinds of ontologies: (i) local ontologies provided by both active and passive entities whose state, properties, capabilities, and services are encoded in, and (ii) the *Sphere Ontology* (SO), which serves as the core of an AS by representing the combined knowledge of all entities. In our approach, we assume that each ATRACO entity (user, agent, device, or service) maintains locally and manages an ontology, which describes its properties, capabilities and current state. In other words, the local ontology of an entity represents the complete set of knowledge associated with this entity. By and large, ATRACO ontologies are divided into:

—*Device and service ontologies*: They are local to each device and proprietary. We assume that they do not follow a standard pattern, because they differently encode the basic characteristics of each device, as well as the features of the services that it provides. Also, they are maintained by the owing device.

—*Policy and task ontologies*: They encode entities and rules that describe specific policies, such as user privacy, multi-user conflict resolution, social intelligence, ontology dissolution, etc. They are considered as part of the NGAIE. The privacy rules, for example, are utilised by a so-called Privacy Manager that takes care of access control and privacy enforcement [KONINGS, B. et al.].

—*User Profile ontologies*: They are constructed around a permanent user profile and more than one temporary profile classes, where user personal information and user specific preferences concerning a specific activity, or service are encoded, respectively. Hence, they encode user traits and preferences, as well as goals, experience, etc. Use can assume different personas based on context.

—*Agent ontologies*: FTA and IA have their own ontologies:
    —*Task ontology*: It is formed by querying the SO regarding the devices and services that are necessary for realizing the tasks in the goal and the task model.
    —*IA ontology*: It is formed by querying the SO regarding the devices and services that offer interactivity, based on the IA interaction policy and the tasks in the goal and the task model.

The knowledge represented by each of the local ontologies is different for each ATRACO entity as it depends on the involved actors' conceptualization (i.e. device manufacturers, software developers, domain experts, ontology experts). On the other hand, this knowledge must be accessible by all ATRACO entities in order to support the semantic interoperability among the heterogeneous local (domain) ontologies, as well as the communication among them. Therefore, a primary requirement is the description and definition of the basic concepts of the ATRACO world model. We have dealt with this issue by developing the ATRACO *Upper Level Ontology* (ULO), which serves as a common semantic referent between ATRACO entities. The ATRACO ULO encodes the basic terms and concepts of the ATRACO model of the world (i.e. devices, users, agents, services, policies) and their interrelations. As local ontologies are treated like black boxes (i.e. they cannot be altered by ATRACO but only by the entities that own them), it is possible that some of them may be highly resource constrained to maintain an ontology or they may maintain only a small set of meta-data describing the services they offer (i.e. UPnP headers). Consequently, the ATRACO ULO is used by the OM in order to optimize the SO.

The SO is formed by applying matching (merging or alignment) algorithms on the local ontologies. It is necessary to combine the information contained in the local ontologies in order to establish interoperability between agents or services using different individual ontologies. Moreover, fresh knowledge can be inferred by combining this information and the AS-related knowledge, therefore can be accessed in an easier way. In order to combine the information contained in the heterogeneous and autonomous local ontologies and reduce semantic heterogeneity among them, we apply ontology matching. Alignment is the result of the ontology matching operation, which finds semantic correspondences between entities belonging to different ontologies. These correspondences can be further used for various tasks, such as query answering, data translation, ontology merging, etc.

ATRACO ontologies are expressed in Web Ontology Language (OWL). In particular, the OWL-DL version is adopted here, which rests on Description Logic and offers the maximum expressiveness, while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). For the purpose of describing services hosted by devices in ATRACO, we have chosen the OWL-S (Ontology Web Language for Services) which provides the means for distinguishing the services in an unambiguous and computer-interpretable form. In order to retrieve data from ontologies defined using OWL-DL, we have chosen an extension of SPARQL, which is a query language for RDF, called as the SPARQL-DL that was introduced in [SIRIN, E. et al.] as a query language for OWL-DL ontologies. As an example, ontology queries, written in SPARQL-DL, were used in order to find which of the devices available in a specific space provide "light" as their service, or to list all the people that are friends of a person "X", etc. The ATRACO system takes user goals and contextual information into account to adapt and reconfigure in a policy-sensitive manner. By combining the above approaches, a totally adaptive system can be developed, as we shall explain in the following sections.

4. ADAPTATION AT NETWORK LEVEL

Several research efforts have addressed the development of an AIE, similar to the ATRACO AmI space, through networking existing devices and resolving interoperability issues with the help of middleware. A standardized approach to Next Generation Networks at home is provided in [Home2015 project] and [Ngn@home]. In [Teaha project] an interoperable middleware having a hardware centric view for creation of universal solutions is proposed, whereas in [Eperspace project] a Home Platform is created to link different devices at home to an interoperable network. In [Amigo project] the aim is to develop ontology based middleware and interoperability. In [Sense project] smart adaptive networks of sensing components are developed. In [ROSE, B. et al.] a home network is described as a set of interconnections between consumer electronic products and systems, enabling remote access and control of those products and systems as well as any available content such as music, video or data, implying fundamental simplicity. Several research efforts have been made to solve the interoperability problem by presenting solutions based on Web Services technology [PERUMAL, T. et al.], connection-based programming [JAHNKE, J. H. et al.], OSGi service gateway architecture [VALTCHEV, D. et al.], [LIN, R.-T. et al.], system core approaches moving the burden of connectivity away from the end devices [BAKER, C. R. et al.], or ubiquitous computing hardware and agent-based service personalization software architectures [DE VICENTE et al.]. Other approaches to modelling and programming devices for AIEs model devices as collections of objects [JAHNKE, J. H. et al.], as Web Services [MATSUURA, K. et al.], and as agents [RAMPARANY, F. et al.]. Clearly, stronger emphasis needs to be put on device adaptation, usability and scalability in order to accommodate new ambient services in a seamless manner. The lack of a de-facto standard middleware for distributed sensor-actuator environments is one of the key issues limiting research on AIEs and their proliferation from research environments to their deployment in our everyday lives. Attempts to adapt heterogeneous environments consisting of dissimilar networks and computing devices primarily address system architectural issues and reconfiguration principles, but broader usability of the developed approaches is limited for AIE applications.

Network Adaptation (NA) is a set of functions and protocols in a SOA middleware allowing an ATRACO system to interact with the network resources. The goal of the NA layer is to provide seamless use of the devices and services to the ATRACO system and to simplify the access to networks in the AIE.

The design and specification process of the NA component involves consideration of several adaptation guidelines, such as unified access on devices that belong to different networks, interoperability between networks, task completion with alternative resources, dynamic discovery for new networks and devices, representation of legacy internet services and relaying of existing Web Services. In ATRACO, NA meets these requirements through defining common device types across networks, allowing events from any network to trigger actions in any other network, featuring a device registry and device/driver manager as well as providing single access for Web Services (e.g. a weather report service) that are able to evaluate availability, accessibility and

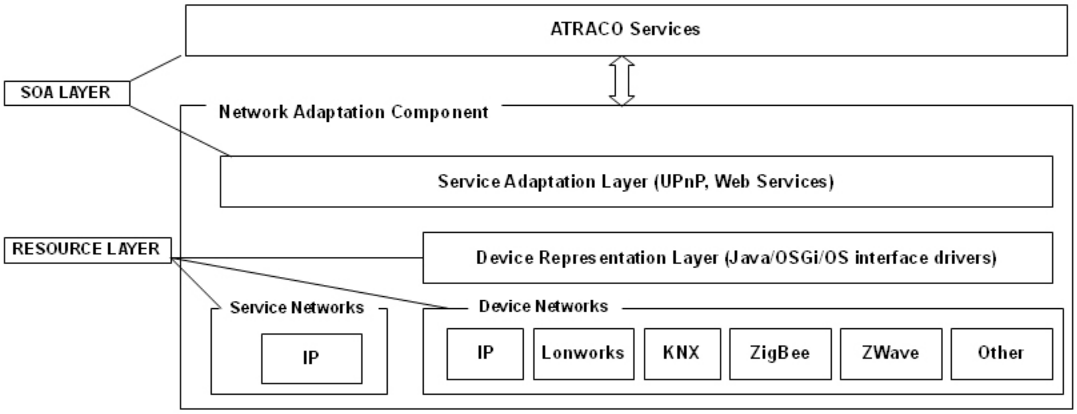


Figure 2: ATRACO Network Adaptation concept.

quality of multiple existing Web Services provided by the internet sites.

Figure 2 illustrates the internal layering of the NA component. To cope with the complexity of accessing diverse control networks, a common Device Representation Layer is introduced. This representation layer is designed in such a way that it handles different networks while hiding the complexity and details of each one of them. For instance, the application, which is to be placed on top of this representation layer, should not tell the difference between a Z-Wave lamp and a LonWorks lamp. Therefore, the device representation layer creates a unique representation of each different device type across networks, which simplifies the evolution of applications and services. In addition to the devices, there are various services which are provided over IP networks. Examples of such services include NTP, VoIP, RTSP, etc. Since these types of services are of interest to an AIE, they can be shaped as ATRACO services and be provided through the NA layer. The functionality in an ATRACO environment is exposed as semantic services which an actor can discover and then compose to form ATRACO applications. After all, each service is associated with at least one semantic description which shields the actor from the complexity of the resource layer realization and makes it easy for the actor to employ these services in accomplishing interesting and useful tasks.

The NA Layer is implemented using a mixed architecture, including a centralized home controller that is able to seamlessly integrate various different devices from different technologies around different home automation and control space together with a variety of modules, some of which may be able to work in a distributed architecture. The Device Representation Layer provides an abstraction of the network devices to OSGi services that can be used by other ATRACO components. These OSGi services represent the functionality of the underlying physical devices, giving the application the opportunity to discover them and retrieve all the required information for managing and performing control actions. The available functional profiles include physical devices with multiple endpoints, binary sensors, level properties like dimmers (with or without timing functionality), direction semantics like shades (with or without timing functionality), direction semantics with read only values, switches, batteries, analog meters, alarms, thermostats etc. Moreover, the Device Representation Layer provides several utilities that intend to provide a service layer for scene execution and alerting. A scene can be considered as a collection of OSGi devices that will execute a number of actions on these devices when a certain event occurs. For instance, an alarm service is included, with which the device representation layer can send alarm events to other OSGi components or higher layers. It makes use of a set of defined alarm levels, types and properties, to cover a wide range of alarm events. Furthermore, the alert service is designed to provide a simple alerting service to the other OSGi aware services in the framework. It listens to the device events and evaluates a series of user defined Rules to finally create the specific alerts.

In the first place, the focus of the Service Adaptation Layer is to provide the Device Representation Layer devices as OSGi services to the other ATRACO entities. On top of that, OSGi device representations are eventually transformed to UPnP services. Another focus of the Service Adaptation Layer is the UPnP adaptation of IP services participating in an ATRACO system. UPnP seems to cover most of the aspects specified for the Service Adaptation Layer as it provides the ease of network setup, device discovery, self-announcement and advertisement of supported capabilities.

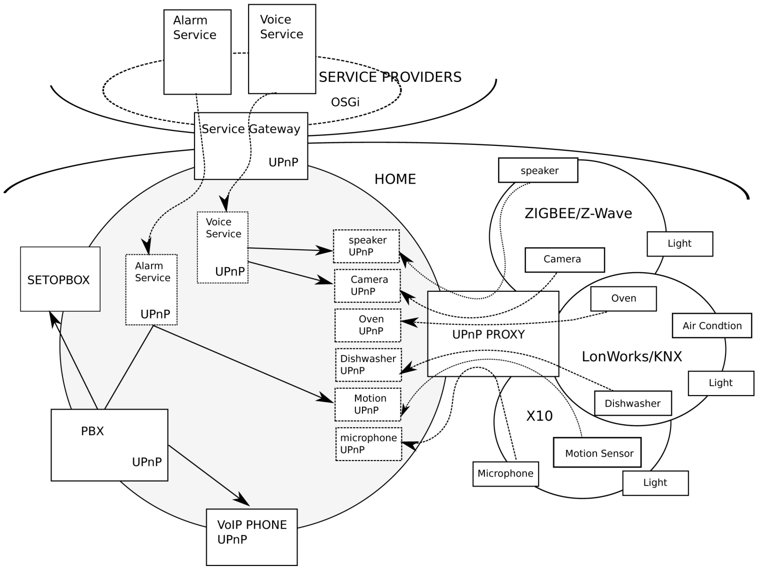


Figure 3:ATRACO generic NA view of an AIE

A UPnP wrapper service would usually track the availability of OSGi devices with certain characteristics and then register a UPnP Device service with the appropriate device and service description XMLs to represent the underlying device. The registered UPnP Device service is then tracked by the UPnP Base Driver and exported to the network as an UPnP device. The NA Layer additionally provides an OSGi based UPnP service for task execution, which is called *Simple Task Execution Module* (STEM). Based on a simple or nested condition evaluation, STEM generates a UPnP event about the result of the evaluation and even executes a set of actions on one or more UPnP devices when the evaluation is true. In order to be able to evaluate conditions, STEM subscribes to the services of the devices that are included in the conditions of the various triggers. STEM, by itself, is not a UPnP control point, so it is not aware of the available devices. It becomes aware of the existing devices using the OSGi API of the OSGi UPnP base driver. When a STEM user adds a scene or a trigger, these are validated with respect to the existing UPnP devices.

NA provides a UPnP virtualization of the interfaced physical devices, which eventually may belong to heterogeneous non IP networks. Figure 3 illustrates a generic view of the networked AIE where UPnP proxies bridge IP networks with non IP networks while representing the devices that belong to non IP networks as UPnP entities. Figure 4 demonstrates the virtualization function of the UPnP proxy. For instance, the proxy knows how to communicate with the ZWave microphone. It uses a special library that encodes -over the ZWave API- microphone commands to start/stop recording as well as to control the gain and the sampling rate. For higher perceptual components, such as the voice recognizer, to make use of the microphone, the proxy represents it as a UPnP device exporting appropriate actions for remote invocation. On the other hand, taking advantage

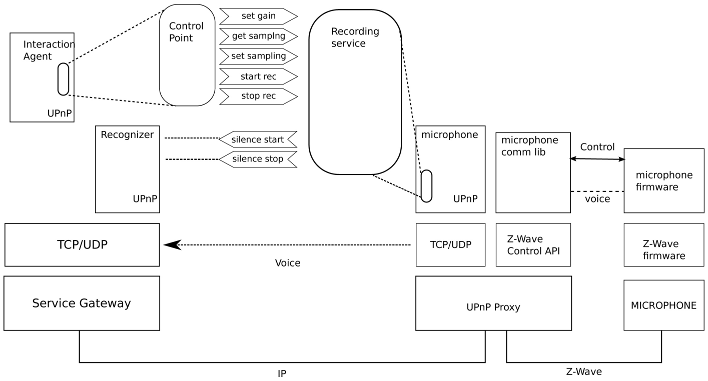


Figure 4: ATRACO NA virtualization function.

of the silence detection feature of the microphone, proxy sends appropriate events which trigger voice processing at the recognizer side accordingly.

5. ADAPTATION AT THE STRUCTURAL LEVEL

There are systems that permit users or agents to aggregate and compose networked devices and services for particular tasks [JAHNKE, J. H. et al. , KUMAR, R. et al. , MATSUURA, K. et al. , RAMPARANY, F. et al.]. However, those devices are not context aware; instead they act more as service providers like Web Services that are usually in the UPnP style. In addition, there has been little work on specifying at a high level of abstraction how such services would work together at the application level taking into account the heterogeneous services, which can be combined in dynamic ways.

In ATRACO an Activity Sphere (AS) is formed to support a user's specific goal and is modelled as an abstract workflow that can be instantiated with the appropriate AIE resources that are available and functional at runtime. An abstract workflow contains a sequence of abstract services that are ontological descriptions of service operations that cannot be directly invoked, but will be resolved during runtime. This is supported by a semantic-based discovery mechanism that is applied to discover suitable services or devices from ontologies that contain the relevant descriptions. Thus an abstract workflow that is defined at design time can be instantiated with the services/devices which exist in a specific AIE at runtime. In this way an AS is dynamic and thus robust to some of the AIE uncertainties, but does require mechanisms that achieve AS adaptation in response to changing context.

In order to achieve the dynamic formation and adaptation of ASs, inspiration has been taken from subtype polymorphism found in the object-oriented programming paradigm. In particular, using a service-oriented approach an AS declares its requirements with the use of concepts that are part of the ontology, instead of the use of concrete resources and devices. With the adoption of ontologies in the deployment of ASs we attempt an extension of the polymorphism mechanism, as the high level concepts in the ontology are mapped in suitable low level resources (e.g. devices) dynamically at the run time. The basic difference with the traditional object oriented models of programming is that the proposed model allows the automatic transformation of type (i.e., type coercion) in a context-aware manner. The binding from high level classes (concepts) into low level classes (devices) takes place by examining the current context, the task that is executed, the policies that have been declared as well as users' preferences. In combination with the *"late binding"* approach of selecting resources at runtime to satisfy specifications from design time, the same AS can be instantiated in different AIEs that contain different resource deployments and context. The different resources that may be involved only need to present a compatible interface to the clients (i.e., in our case, a UPnP interface). This dynamic binding is therefore dependent on the context in which the binding occurs.

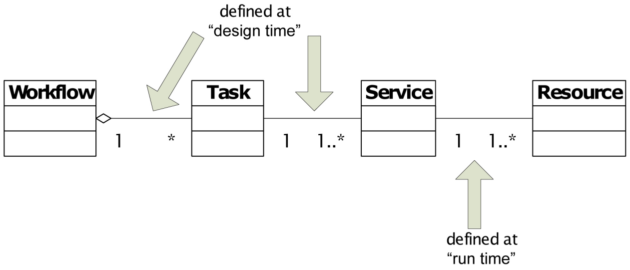


Figure 5:Conceptual model for Dynamic Service Binding

Figure 5 demonstrates a conceptual view of the model behind the adaptation at the structural level. The plan to achieve a user goal as an execution path of services is represented as a workflow. A workflow is mapped into a number of tasks and a workflow task is mapped into one or more services. In addition, each service would also require certain physical resources for its implementation. Mapping of the task to abstract services can be specified at design time with regards to the users' functional requirements. However, mapping of the abstract service to the actual human and physical resources is done at runtime in order to keep up with service orientation.

ATRACO-BPEL, a streamlined version of BPEL (Business Process Execution Language) [Web services] is used to specify the plan of activities that serves a goal as a workflow of device and agent services. While BPEL is a suitable language for describing workflows, an ATRACO workflow description presents requirements that cannot be completely covered by BPEL. This is due to the fact that BPEL partners (partnerLinks) are statically bound to specific Web Services. In the context of ATRACO, however, services are not bound at design time; instead, they are dynamically bound during the execution of the workflow. Thus, there is a need to describe services in the workflow by their semantics which mainly define search terms related to ontologies (e.g., "Luminosity" for a light service). In the following example, we explain how the ATRACO-BPEL is used for a task that is bound with the appropriate service(s). The task *AdjustLights* is associated with the partnerLink *AdjustLightsPL* as part of the orchestration logic section.

```
<bpel:invoke name="AdjustLights" partnerLink="AdjustLightsPL"></bpel:invoke>
```

The partnerLink *AdjustLightsPL* has an input role (myRole) called *ATRACO:lightStatus* and an output role (partnerRole) called *ATRACO:triggerLight*. The Continuous type denotes that the execution of the activity is to be treated as a task that is running continuously, i.e., the workflow does not wait for its termination.

```
<bpel:partnerLink
        name="AdjustLightsPL" partnerLinkType="ATRACO:Continuous"
        myRole="ATRACO:lightStatus" partnerRole="ATRACO:triggerLight ">
</bpel:partnerLink>
```

In the following we explain only the output role which denotes the appropriate abstract services (*Light or Luminocity or Dimmers*) that must be bound to fulfil the role along with any other application specific details that are needed for its operation (i.e., the task will be monitored by an ATRACO FTA agent for learning user behaviour with respect to light adjustments and all found light or dimmer devices are to be used).

```
<ATRACO:role
        name=“triggerLight” type=“output” FTA=“yes” IAmode =“withFTA”>
        <ATRACO:service semantics= “Light;Luminosity” selector = “all” precond = “”>
        </ATRACO:service>
        <ATRACO:service semantics= “Dimmer Control” selector = “all” precond = “”>
        </ATRACO:service>
</ATRACO:role>
```

Each *ATRACO:role* envelops a set of services that are bound to it. Each role can have more than one abstract services. For each abstract service specific attributes are defined, providing the necessary support for device discovery and service operation. In the example above, the output role definition will lead to the discovery of all lamp devices and dimmer controls. It should be noted that the IAMode setting refers to the Interaction Agent (*withFTA* value specifies that there is a need to find proper user inputs for the FTA monitored tasks) and there are no special constraints to be met prior binding the corresponding device(s). The ATRACO software component that manages the AS adaptation at the structural level is the Sphere Manager (SM) which has a role to dynamically bind devices found in an AIE by taking into account the abstract workflow, the availability of resources (e.g., devices, networks and related services) and contextual information. This is achieved in collaboration with the Ontology Manager (OM) component which provides service descriptions, personal and contextual information at the time of binding. The OM administrates a knowledge repository, by performing ontology updating, querying and aligning services, to support the operation of an AS. The SM is responsible for executing the workflow while successfully preserving the precedence constraints or the conditions that are specified in the workflow and adapting the workflow by rebinding services to alternative devices in case of device failure and/or when new devices enter the local network (handling of adaptation events).

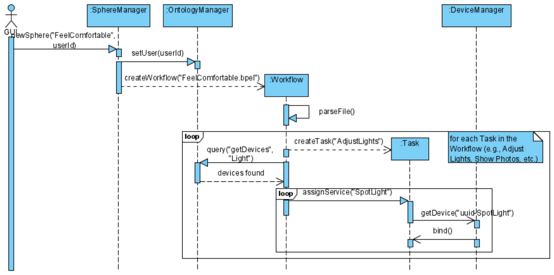


Figure 6: “Feel Comfortable” Activity Sphere instantiation and binding of devices to services

The sequence diagram in Figure 6 shows the basic interaction of the software components during the instantiation of an AS called “Feel Comfortable”, which employs the dynamic service binding process described above (methods used inside the two loops). The Workflow object aggregates a number of Task objects where each of them represent an activity in the workflow (i.e., “Adjust Lights”). The services required to run this activity are divided into input and output services and are connected with the appropriate resources. The resources that are bound to the Task object can be either devices that the Task directly controls (i.e., input sensors and actuation devices) or agents, such as the IA or the FTA. In either case, the Task object is informed about the status of the resource and operates according to the pattern specified by its type.

While we share many common features with other composition systems [BECKER, C. et al.], [RANGANATHAN, A. et al.], [VERMA, K. et al.], such as automatic discovery and dynamic binding of services, the use of ontologies to access semantic descriptions of services and to reason on the appropriateness of a device, our ontology-based services composition framework supports characteristics that are not found in the mentioned systems. Composition and re-composition of the abstract form of an application is automatically generated by the planning process [BIDOT, J. et al.]. This means that modifications in the model of the application in [BECKER, C. et al.] and [RANGANATHAN, A. et al.] must be performed by the developer whereas in our case, this is facilitated by using AI planning techniques to produce abstract service workflows in a more structured and efficient way. This allows for the efficient handling of events such as when the user's task intention changes drastically. As in the case in [RANGANATHAN, A. et al.], service failure recovery and adaptation due to mobility, while conforming to the given abstract description, is done at run-time through an ontology-supported service binding mechanism. However, the use of ontology alignment (outlined in Section 7) allows our system to accommodate heterogeneous device specifications that may be different at a lexical and structural level with a degree of accuracy. In addition, by modelling agents as services, we can incorporate higher-order adaptation semantics into our service composition framework, either in the form of FTAs, which learn user preferences using fuzzy-based linguistic models, or as IAs.

6. ADAPTATION AT THE BEHAVIOURAL LEVEL

User Behaviour Adaptation (UBA) is one of the core dimensions of adaptation which is implied in the AmI vision to realise AIEs. UBA is the defining term for the adaptation of the intelligent systems within an AIE to the users' preferences and desires which are subject to change with regards to a large variety of factors (time, mood, weather, etc.). The need for UBA arises from a series of requirements that AIEs are faced with, specifically the ones for transparency and intelligence [DUCATEL, K. et al.]. In AIEs, transparency is characterized by the seamless and "invisible" integration of the devices into the surrounding environment or, in other words, the functionality provided and encapsulated by the environment should be provided in a seamless and non-intrusive manner. This vision of a non-intrusive AIE has been described in detail in [RIVA, G. et al.]. Furthermore, the AIE should be able to transparently learn from the user and the way he/she interacts with his/her environment in order to anticipate the user's preferences in future situations and to maintain an environment which correlates with the user's expectations as much as possible at any time. While there are other ways of realising similar functionality (such as pre-programming the (devices in the) environment), the challenge is the creation of an environment which continuously learns from the user and adapts to his/her changing preferences without generating demands on the user but by simply learning from his/her natural interaction with the environment.

Realising UBA in AIEs is quite challenging as creating complex mathematical models of dynamic real world systems is unrealistic and the necessary mapping of sensory information onto an intelligent and pre-emptive set of behaviours is too complex to be precompiled and hardwired into the system. Besides, in order to manage the multitude of computational devices and artefacts available in our living spaces, there is a need to employ a transparent and distributed layer of intelligence that can be personalized with regards to the user(s)'s needs and preferences by learning from their behaviour and thus helping the user(s) to configure and control their environments. The UBA should be implemented using techniques that have low computational overheads to effectively operate on the embedded hardware platforms present in the everyday environments which have small memory and processor capabilities. In addition, the employed UBA 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 life-long learning mechanism that will allow the system to adapt to the changing environment and user preferences over short and long term intervals. In all cases, it is important that these intelligent approaches represent their learnt decisions in a form that can be easily interpreted and analysed by the end users. There is also a need to provide robust mechanisms that will allow handling the various forms of uncertainties so that the system operates under the varying and unpredictable conditions associated with the dynamic environments and the user preferences. Such uncertainties can be categorized in two classes as follows: i) the environmental uncertainties; ii) the user uncertainties.

Several projects have tried to realise UBA within AIEs. In Sweden, Davidsson [DAVIDSSON, P. et al.] utilised multi-agent principles to control building services. In Colorado, Mozer [MOZER, M. et al.] used a soft computing approach based on neural networks which focused on the intelligent control of lighting within a building. Work at MIT in the HAL project [COEN, M. et al.] concentrated on making a room responsive to the occupant by adding intelligent sensors to the user interfaces whereas Context Aware systems are the focus of the Aware Home work at Georgia Tech [Abowd, G. et al.]. These projects represent a large body of current research effort; however, they are mostly concerned with time independent context rather than temporal history, learning or adaptation, which all together are central to our requirements for agents supporting the AmI vision. There are other high profile intelligent environments projects such as the Microsoft Smart House [BRIAN, B. B. et al.], BT's Tele-care [SHERWIN, A. et al.] and Cisco's Internet Home [Internet protocols]. However, most of these industrial projects including home automation technologies, like Lonworks and X10 are geared toward using networks and remote access with some autonomous control. In general, this is mostly simple automation with sparse use of Artificial Intelligence (AI) and little emphasis on learning and adaptation to the user's behaviour. Some researchers have employed fuzzy systems for UBA like [RUTISHAUSER, U. et al.] and [DOCTOR, F. et al.]; however, these systems learnt only the user rules and not all the parameters needed for UBA.

From the above discussion, it is obvious that no previous work has managed to realise the full requirements of UBA in AIEs where intelligent systems will be able to learn the user behaviour and model it by monitoring the user actions. There was a need also for systems that could create models which could be evolved and adapted online in a life learning mode. These models need to be transparent and easy to be ready and interpreted via the normal user in order to enable the user to better analyze the system and its performance. These UBA systems could be used to control the environment on behalf of the user and according to his/her satisfaction to perform the given tasks.

UBA within ATRACO is responsible for the learning about, as well as the adaptation to, the user in the respective environment. In particular, it focuses on short and long term adaptation to the changing behaviour of the user in order to fulfil a given task. In order to achieve this, UBA relies heavily on the interaction of the user with his/her environment and the devices contained. This allows the UBA to maintain the vision of a transparent AIE by avoiding any additional load on the user which would, for example, be created by a requirement for programming or "teaching" of the ATRACO system. Within ATRACO, we employ intelligent agents to realise UBA which can adapt the operation of the AIE, to the user. The very basis for an intelligent UBA agent is the ability to learn (about) the user behaviour. In practical terms and in the light of ATRACO, this can be reasoned to mean the learning of the users' preferences in terms of the states of a specific set of devices, which are part of a specific task. While learning about the user behaviour is essential, it is clear that it is not sufficient. A realistically useful, intelligent UBA agent should be more than an automation of the users' recorded preferences; indeed, it should be able to adapt to changes of the users' preferences on the fly, to update and modify what it has learnt as and while the user continues to live in the environment. Short term user behaviour adaptation designates the ability of the UBA agent to adapt its operation to short term changes in the user behaviour with respect to the user preferences. In other words, the agent needs to be able to update its learnt knowledge about the user as and when it is gathering new information about the user preferences which can either be:

—Information about a new aspect of the user behaviour (e.g., the user starts reading in the morning instead of watching TV, thus turning the reading light on).

—Updated information about a previously seen user behaviour (e.g., the user wants the reading light be dimmed to about 80

—Contradictory information about a previously seen user behaviour (e.g., the user decides to read in bed, therefore requires his/her bedside lamp to be on instead of his/her reading light).

On the other hand, long term user behaviour adaptation designates the adaptation of the learnt user behaviours over longer terms, i.e. weeks, months and years. Specifically, the UBA agent should be able to adapt to changes in the user behaviour which are related to variations in the user preferences (i.e., the users habits vary over time, e.g., he/she might go to bed earlier as he/she ages), environmental changes (e.g., during the winter, lower intensity of the sun leads to less ambient temperature within the AIE, which in turn might result in higher demands on heating) as well as combinations of the two (e.g., less sunlight in the morning during the winter might mean more lighting is to be demanded by the user). As such, the UBA agent should be able to maintain knowledge over long periods of time while aiming to maximise its ability to generalize (e.g., in adjusting the lighting levels for the user in the morning) while also serving the users' specific requirements across long time spans and different environments.

Within ATRACO, one major focus is the research regarding the adaptation to the behaviour of not an individual user, but multiple people. While the presence of a single user presents the UBA with a large number of intra-user uncertainties - i.e., the variation in a user's preferences and behaviour over time; the presence of multiple users presents the UBA with inter-user uncertainties, which arise from the combination of multiple users' preferences that are to be satisfied at the same time. It is clear from human experience that this is usually not possible and that either a compromise needs to be found, or the preferences of one (or none) of the individual users need to be selected. Within ATRACO, the UBA is aiming to deal with the presence of multiple users and as such to address the complexity of inter-user uncertainty which has largely been ignored in the field of AIE.

As the UBA agent learns from the user in order to - it could be argued - program the environment on his/her behalf, it is very important to preserve the interpretability of the UBA agent's actions. Traditionally, rule based expert systems have frequently been used in order to provide reasoning capabilities while interacting with a human user [REMAGNINO, P. et al.]. The main advantages of rule based expert systems are their reliability and interpretability. Fuzzy Logic Systems (FLSs) have been shown to have great potential at dealing with uncertain information in a large variety of contexts from industrial control to AIEs [RUTISHAUSER, U. et al.], [DOCTOR, F. et al.], [HAGRAS, H. et al. 2007], [HAGRAS, H. et al.]. The nature of FLSs, which are based on fuzzy set theory (an extension of classical, crisp logic theory), make them - in our opinion - ideally suited as a basis for the development of an interpretable, resilient, flexible UBA. With fuzzy logic systems being a well-established field within artificial intelligence, we will not describe the basic notions of fuzzy logic theory here (the interested reader can find a large variety of books and scientific papers, including [MENDEL, J. et al.] and [YEN, J. et al.]). The requirement for learning of the UBA is addressed by employing fuzzy systems methodologies, specifically by relying on fuzzy logic rule based systems. Similar to the traditional rule based systems, fuzzy rule based systems are based on logical rules in the form of:

IF X is a AND Y is b AND Z is c THEN O is s

, where X, Y and Z are inputs or variables, a, b and c are states of these variables, O is an output (or variable) and s is a state of O. X, Y, Z and O are fuzzy sets which encompass the uncertainty and fuzziness associated with the input and output variables in the fuzzy rules. The simple structure of these rules allows for straightforward interpretability by either the user or the expert designing or maintaining the system. The potential of fuzzy rule based systems lies in the design of both the input and output variables, and, most importantly, in the number of active rules and their respective antecedent/consequent matching. In order to generate these fuzzy rules, which effectively consist of the extracted or learnt knowledge about the user's interaction with the environment, we follow a 3-step approach:

—In the **first step**, the user's interactions with the environment are observed and logged.

—In the **second step**, fuzzy logic sets which encompass the uncertainty associated with the various devices are constructed from the raw data records.

—During the **third step**, the data collected during Step 1 is combined with the fuzzy sets created during Step 2 in order to create a series of fuzzy rules that reflect the higher order knowledge which can be gathered from the user-environment interaction.

Within the UBA, this extraction of fuzzy rules is based on methods highlighted in [DOCTOR, F. et al., HAGRAS, H. et al. 2007]. The essence of the rule extraction system is the extraction of a *manageable set* of fuzzy rules which in general reflects the very large number of logged user-environment interactions as good as possible. The number of rules we refer to as manageable is largely dependent on the following factors:

—The processing resources.

—The requirement on interpretability (i.e., as the number of rules increases, it becomes more and more complicated to interpret the UBA functionality).

As the employed method naturally groups similar user-environment interactions (e.g., the user turns his dimmable reading lamp to 80% one day and to 85% the other day (due to differing outside light conditions)), it provides a natural way for generalization - an essential ability for the emergence of intelligence. After the completion of Step 3, the UBA should have generated a set of fuzzy rules that reflect the user-environment interaction which was recorded during the learning phase. For the UBA to continuously adapt to the users' changing preferences and behaviour, this current set of fuzzy rules requires constant adaptation.

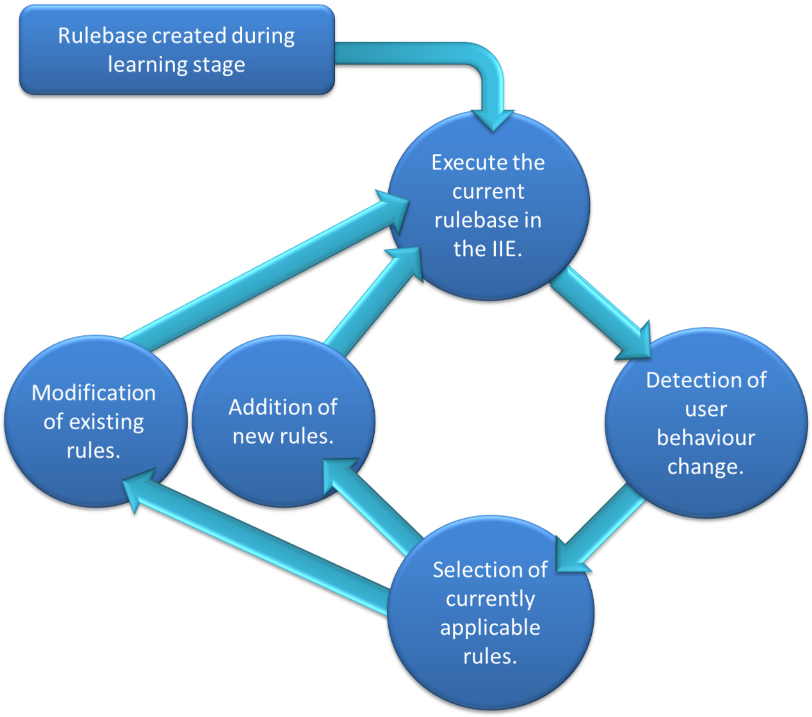


Figure 7: Short term adaptation.

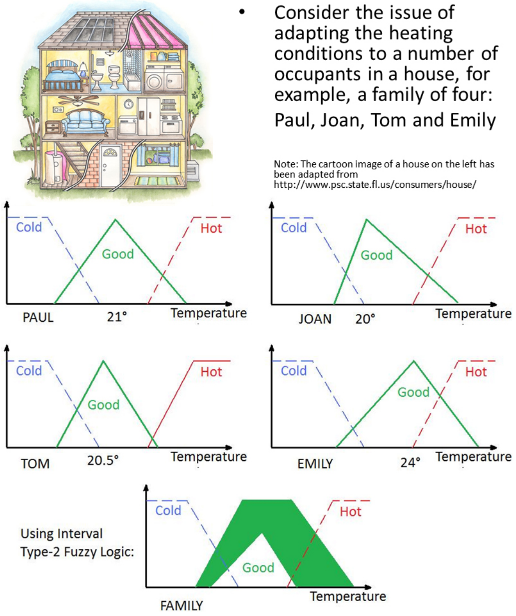
Short term user behaviour adaptation refers to the adaptation of the agent to the short term changes in the user behaviour, i.e., changes that occur within hours or days. Within ATRACO, short term UBA is handled using a rule updating approach introduced by Doctor et al. in [DOCTOR, F. et al.]. The rule adaptation algorithm is summarized in Figure 7. By adjusting the learnt rule base unobtrusively on-the-fly, the UBA aims to adapt to the user in a rapid and transparent fashion, maintaining the "user-is-king" paradigm. While the adaptation over short periods of time is described above, some changes only occur slowly over long periods of time. Long term adaptation refers to the adaptation to changes in the user behaviour introduced over long periods (e.g., the gradually changing preference for brighter lighting indoors as the mornings become darker and darker during autumn, or the preference for warmer indoor temperature as a person ages).

Within ATRACO, long term UBA is achieved by a two-fold strategy:

—zSlices based general type-2 fuzzy sets, introduced in [WAGNER, C. et al.], which are employed to capture limited variations in the user preference/behaviour. As such, type-2 fuzzy rule bases are able to provide more generalised rules which can provide superior adaptation results over long periods of time [HAGRAS, H. et al. 2007].

—For long periods of time, a review of the capture information about the user-environment interaction in order to produce new rules based on type-2 fuzzy sets using the method reported in [HAGRAS, H. et al. 2007].

The interpretability of the UBA is maintained throughout the adaptation stages through the use of fuzzy rules where the nature of fuzzy logic rule based systems allows for direct interpretability through the use of human-readable rules. Within ATRACO, the complexity of the information in the rules is largely "hidden" within the nature of the fuzzy sets which can be of type-1, interval type-2 or zSlices based general type-2 fuzzy systems. This is well illustrated in the example below (Figure 8) detailing the role of zSlices based general type-2 fuzzy logic systems in order to support the presence of multiple users with an intelligent inhabited environment (IIE). The presence of multiple users within an environment is handled through the use of zSlices based general type-2 fuzzy sets and systems which allow the amalgamation of multiple single-user adapted preferences/behaviours on the fly. This is achieved by representing agreement between multiple users over multiple zLevels as illustrated in the example shown in Figure 8. The figure presents the example of a given AIE environment occupied by a family of four, where each member of the family has a slightly different preference in terms of the indoor temperature. This is visualised using type-1 fuzzy sets, and it can be seen that the type-1 fuzzy sets for good temperature are located at different locations for each family member.



- Consider the issue of adapting the heating conditions to a number of occupants in a house, for example, a family of four: Paul, Joan, Tom and Emily

Note: The cartoon image of a house on the left has been adapted from http://www.psc.state.fl.us/consumers/house/

Figure 8: A Conceptual example of multi-people UBA.

One method of addressing the presence of multiple family members is the use of interval type-2 fuzzy logic sets as shown in Figure 8. This allows for the capturing of all the family members' preferences without requiring additional rules. (Basically, a virtual person which encompasses the preferences of all family members is created.) Nevertheless, it is equally visible from Figure 8 that while Joan, Tom, and Paul all prefer an indoor temperature around 20.5 degrees Celsius, the choice of Emily is different from those as she prefers a temperature of 24 degrees. zSlices based general type-2 fuzzy sets can be employed to model the agreement between Joan, Tom, and Paul while still reflecting Emily's preference. They can be seen as a weighting of behaviours/preferences according to the level of agreement, again, without requiring the creation of additional rules, thus supporting good interpretability.

In the ATRACO architecture the component that provides the UBA functionality is the Fuzzy logic based Task Agent (FTA). The ATRACO system initializes one or more FTAs depending on the task workflow complexity associated with an Activity Sphere.

7. ADAPTATION AT THE SEMANTIC LEVEL

Semantic adaptation refers to changes in order to deal with any disturbance that would affect the meaning associated with an AS. For example, the user's goal of feeling comfortable has the same meaning for the user, according to the user's preferences, but independently of the NGAIE in which it will be realized (i.e., at home, or at a hotel room) and of the AE resources that have to be available in order for the user to be satisfied. Indeed, one of the main problems while migrating AS across NGAIEs is the different kinds of resources that may be available in the two environments. Still, an AS has to pursue the achievement of the goal when changes on the type of the available resources occur (as agents and users may come and go, devices and services may appear and disappear in time) and when changes on the cardinality of the available resources occur (as the numbers of devices, or users that participate in the realization of an AS may differ in time).

Our system will adapt an AS by discovering equivalent resources of an AE that can perform the same task in order to ensure an uninterrupted execution of the AS, while shifting from one NGAIE to another. In ATRACO, each of these resources carries its own ontology. These ontologies are heterogeneous and may be also inconsistent since they are developed independently by device manufacturers, software developers, or non-domain experts. Therefore, in order for the applications to properly operate in such a dynamic environment, a solution must be found to the semantic heterogeneity problem, which results from the co-existence of multiple individual heterogeneous ontologies.

Ontology alignment is a solution to the aforementioned problem. Establishing semantic links, that is, alignments, between these ontologies is a necessary precondition to achieve interoperability between agents, services, or applications using different individual ontologies. These alignments can then be used in order to translate requests and data between ontologies, to evolve the ontologies, or to merge the ontologies. The alignment task can be seen as a function [BOUQUET, P. et al.], which returns an alignment between a pair of ontologies by making use of a set of parameters concerning the matching process (i.e., matching algorithms, user involvement), and a set of available resources (i.e., online ontologies, predefined alignments).

Past attempts to ensure interoperability among context elements within a ubiquitous computing environment can be classified into three alternative approaches, which can be used in combination:

—A priori standardization of comprehensive and well-formed ontologies: These ontologies are usually divided into an upper ontology and several domain-specific ontologies. The upper level ontology captures general context knowledge about the AmI environment and defines the basic concepts of person, location, time, activity, computing entity, etc. The domain specific ontologies define the details of general concepts and their properties in each sub-domain where they apply to. Projects SOCAM [GU, T. et al.] and GLOSS [DEARLE, A. et al.] follow this approach. A few approaches [ALEKSOVSKI, Z. et al., ALEKSOVSKI, Z. et al. 2006 and SABOU, D. et al.] have also considered the use of external knowledge as a way to obtain semantic matching between dissimilar ontologies. Although the use of ontologies may indeed favour semantic interoperability, it relies on the existence of agreed domain ontologies in the first place. Furthermore, these ontologies will have to be as complete and as stable for a domain as possible, because different versions only introduce more semantic heterogeneity. Thus, semantic-integration approaches based on a priori common domain ontologies may be useful for clearly delimited and stable domains, but they are untenable and even undesirable in highly distributed and dynamic environments.

—A priori standardization of matching links among consistent ontologies: Matching links between two ontologies could be available either by application developers or even end users. These matching links or correspondences can be used, for example, by agents, who do not share the same content language and ontology, in order to understand each other's messages. This is the approach followed by the GAIA [ROMAN, M. et al.] and CobrA [KAGAL, L. et al.] projects. However, this approach does not take into account the contemporary evolution of ontologies and their respective alignments.

—An online available ontology alignment service among well-formed ontologies: This service matches two ontologies upon request. In [EUZENAT, J. et al. 2008] an ontology alignment service is used in order to help agents in finding an alignment between different ontologies they face. This service provides mechanisms for: retrieving past alignments; matching two ontologies dynamically; translating queries as well as answers to queries between context managers that use different ontologies; and finding out ontology close to a specific ontology. However, the authors do not consider how an ontology alignment service can be applied in the case of inconsistent and imperfect ontologies, which is our concern.

In order to bring the ATRACO domain ontologies into mutual agreement, the Alignment API [EUZENAT, J. et al.] was chosen to be the core of our ontology alignment solution as it provides an implementation of a general infrastructure that can be enriched by additional algorithms, ontology matching formalisms support and interfaces. The implementation based on the Alignment API allows to: select two ontologies for alignment via providing their web URIs, select an alignment method among the built-in ones and instantiate the alignment process; browse through the proposals of the algorithm for ontology alignment and choose the acceptable ones; and save the chosen correspondences in a common repository available for reusing. In the case where a pair of non-overlapping ontologies cannot be directly aligned automatically, as well as in the case where an evaluation of the produced alignments is required, the alignment process relies on a "trusted third party". External ontological resources, as existing ontologies sharing a similar context with the aligned ones, either the WordNet, manual intervention, or specialized agents play this role, in the above cases.

The approach in implementing our alignment solution uses a combination of terminological approaches, which exploit string similarity between labels of ontologies with structural approaches, which rely on the structure of the ontologies. For assessing lexical similarity of concepts and properties, we use the Levenshtein edit distance metric between two strings, enhanced with a synonyms database that we maintain (an external resource such as WordNet could also be used). Levenshtein distance is defined by the minimum number of operations needed to transform one string into another, where an operation may be an insertion, deletion, or substitution of a single character. However, lexical similarity is not always adequate to verify that concepts are semantically compatible. We, therefore, investigate whether their direct ancestors and descendants as well as sibling entities share lexical similarity. In this way, we can assess the structural locality of concepts. However, the approach for the alignment has its limitations. Satisfactory results are obtained when the input ontologies are in the same domain, and furthermore, when non taxonomic relationships (i.e., restrictions) between concepts are not taken into account.

In the ATRACO architecture the component that provides ontology management and aligning

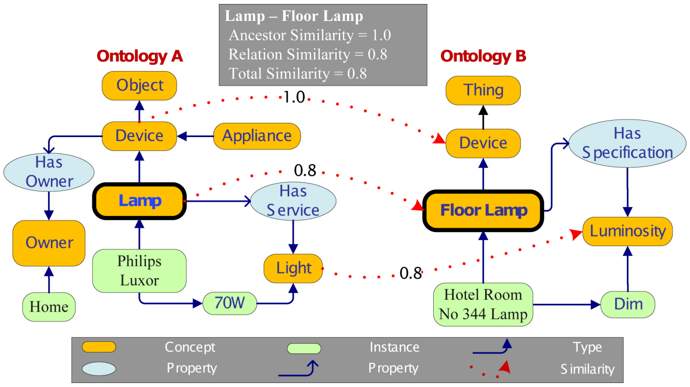


Figure 9: An example of ontology alignment

services is the Ontology Manager (OM). OM has been developed as a wrapper around the Jena Framework, and its interface supports querying the ontology using SPARQL. The output of the alignment process is an OWL ontology that includes OWL class equivalence assertions that relate the equivalent concepts to the input ontologies. After the alignment, inference and querying are performed on a grid of imported ontologies, given the alignment points that have been produced. An example of alignment is shown in Figure 9. The example implies that the realization of the user comfort at a hotel room with respect to the need of using a light service is possible, as at home, since the floor lamp of the hotel can be used instead of the home lamp. The ontologies of these devices are aligned and the system can infer that they are providing similar services, despite their inconsistent descriptions at lexical and syntactical level.

8. ADAPTATION AT THE HCI LEVEL

For a few years, access to computers has become possible to a large variety of users (kids, seniors, novices, disabled people, etc.). At the same time, advances in the miniaturization of electronic components have allowed the development of a large variety of portable devices (mobile phones, portable media players, personnel digital assistants, etc.). New interaction situations have started to appear due to users' mobility enabled by this evolution. It is nowadays common to make a phone call on the street, to work in public transportation while commuting, or to read e-mails at a fast-food restaurant. Furthermore, with AEs, the interaction environment which was static and closed becomes open and dynamic, instead. This variety of users, systems and physical environments lead to a more complex interaction context. It is no longer reasonable to continue to propose static and rigid interfaces while users, systems and environments are more and more heterogeneous. To the heterogeneous character of the interaction context, the user interface must respond by a dynamic adaptation to preserve its utility and usability.

Most approaches that are used to resolve the problem of user interaction heterogeneity consist of trying to modify (adapt) the behaviour of the user interface depending on the interaction context. The main differences between these different approaches concern the way they achieve this adaptive behaviour and in particular how they manage questions such as: When is the adaptation done? Which components are adapted? How are they adapted? By who are they adapted? In fact, adaptation can be characterized using different criteria depending on the adopted point of view (user centred [DIETERICH, H. et al.], adaptation target oriented [BRUSILOVSKY, P. et al., KOBSA, A. et al., and STEPHANIDIS, C. et al.], software architecture [THEVENIN, D. et al.], etc.). Among the different points of view, we can emphasise the five following criteria:

—Actor: represents the entity that is responsible for the adaptation task. It can be the user, the designer, the system, etc.

—Components: represent the software entities that will be modified to achieve adaptation. It can be the help system, the kernel core, the task model, the dialog controller, the physical or logical interaction objects, etc.

—Time: represents the moment when the adaptation is performed. The adaptation can be static (performed during the design phase), dynamic (performed at runtime), or sometimes performed between sessions. Certain authors refer to static adaptation by the word "adaptability" while the dynamic adaptation is referred by "adaptivity" [FRASINCAR, F. et al.].

—Direction: represents the adaptation orientation. The system may adapt its outputs and/or adapt itself to inputs (artefacts adaptation).

—Target: represents the entity towards which we want to adapt. This is usually denoted by "interaction context".

Several definitions exist to describe the notion of interaction context. Within the HCI research community, the most used definition is the triplet <User, System, Environment>[CALVARY, G. et al., SAMAAN, K. et al., and VANDERDONCKT, J. et al.]:

—User: is described by a profile which informs about its preferences, cultural characteristics, cognitive and sensory-motor capacities, etc. Those can be static (such as a permanent handicap) and/or dynamic (e.g., the user is not looking at the screen).

—System: represents the physical (devices) and logical (software) resources.

—Environment: represents the physical environment where the interaction is done (luminosity, noise level, etc.)

Certain authors include some other information such as the current user activity [TARPIN-BERNARD, F. et al.] (which can be attached to the user profile) or the time, season, and temperature (which can be attached to the environment). To resolve the problem of interaction adaptation, different kind of approaches exist: translation approaches [FLORINS, M. et al.], retro-engineering and migrating approaches [BERTI, S. et al., PAGANELLI, L. et al.], Mark-up language based interfaces and model based interfaces [EISENSTEIN, J. et al., STANCIULESCU , A. et al.]. Most of these approaches follow a top-down procedure and try to design several interfaces for the same application, each one being adapted to a specific interaction context. This kind of approaches can be used when the different possible interaction contexts are not very numerous and well identified.

In ATRACO (and in AIEs in general), the context is very rich and highly dynamic. For instance, available devices and services may be discovered and may change dynamically at runtime. Hence, the previous approaches are not well suited to our needs. We adopt another approach which consists of integrating the adaptation mechanisms inside the system itself thereby allowing the interface to modify its behaviour dynamically in order to stay pertinent with regards to the interaction context. This kind of approach is better suited to our needs as the interaction contexts in AIEs may vary strongly. In ATRACO, the module that is in charge of these adaptation mechanisms is called Interaction Agent (IA).

Thanks to the interaction richness it can offer, multimodal interaction represents an interesting solution to the User *Interaction Adaptation* (UIA) problem. Hence, the IA integrates a knowledge-based system that addresses three levels of adaptation: *allocation* which is the problem of selecting the modalities and devices through which the user interface will be expressed; *instantiation* which is the issue of selecting the appropriate parameters for an allocated set of modalities and *evolution* which addresses the evolution of the user interface during interaction. Furthermore, we model the context as an Ontology called the *Interaction Ontology* (IO) which describes the different pertinent elements of a user interface - from the user to the device - along with other information about the context. By using ontology, we can adapt the same model to various environments by aligning the concepts in the IO with instances from the rest of the Sphere Ontology (SO). The IO structure is tightly bound to the IA implementation; it represents what the IA can understand from an environment. This way, the IA can make requests to the OM, based on the vocabulary it can understand from the IO, and retrieve context specific information from the SO. We thus have a model of interaction context that is consistent and conceptually equivalent from one environment to another.

As we have discussed before, we need to free ourselves of the fully top-down approach which is not suited to highly dynamic interaction contexts. However, as a fully bottom-up approach would lead to inconsistencies in the flow of interaction tasks [SCAPIN, D. et al.], we have opted for a mixed approach in which a set of already existing interaction objects are combined to generate a user interface. We call those interaction objects *Off-the-shelf Interaction Objects* (OIO) because they are already implemented in the programs that we reuse and recompose at runtime. Each of those OIOs can understand different interaction languages and be manipulated by different interaction techniques. For the moment, the OIOs that we dispose of include two types: graphical user interface and complex natural dialogue descriptions. These OIOs are of course described in the IO, so that the IA is able to infer the devices that are needed to instantiate them. The aim of the IA is then to select the right couple <OIOs, devices>depending on the context. Such a couple is called Final Interaction Object (FIO). An OIO can be a simple button, or it can be a whole widget for playlist edition. This way, designers have a total freedom on the internal behaviour of the OIO, which lets them to implement new and/or rich interaction techniques. Indeed, one of the drawbacks of automatic generation or computer assisted generation of user interfaces is that the outcome is often monotonous and offers poor interaction capabilities whereas hand-crafted user interfaces may offer richer and more appealing interactions. Hence, our approach allows for both hand-crafted user interfaces and automatically generated interfaces to coexist. The former will generally refer to complex tasks, and the latter could be used for simpler tasks.

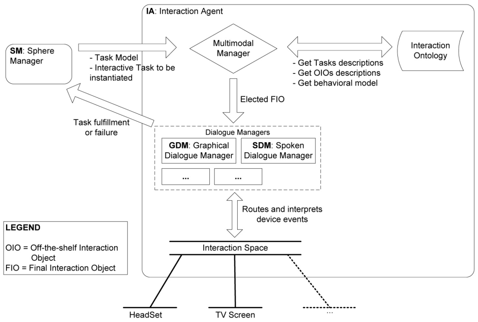


Figure 10: Interaction Agent architectural overview

Figure 10 represents the internal structure of the IA. The main module is the Multimodal Manager (MM). It is responsible for deciding which modalities should be instantiated by the use of which devices and how - i.e., allocation and instantiation. The MM furthermore addresses these issues in a continuous way during an on-going task. In this manner, it supports evolution. The IO is the knowledge base that encodes the current state of the context and the behavioural model. For controlling tasks (e.g., switching the light on), it is feasible to automatically allocate, instantiate and evolve the user interface. But, when it comes to more sophisticated tasks such as human computer *negotiation*, linguistic modalities are necessary because they offer the feature to dynamically generate rich semantic content thereby proposing a more natural way of interaction. Moreover, an exploratory study [BELLIK, Y. et al.] showed that users tend to prefer using the speech modality within ambient systems. For these reasons, we have developed a separate module, called the SDM (Spoken Dialogue Manager), which is dedicated to provide a coherent rich spoken dialogue that enables the evolution of an on-going dialogue.

9. PROTOTYPE

In order to test the proposed architecture, we have implemented a prototype of an ATRACO system. It consists of the components detailed in the previous sections and several basic components for controlling the environments (e.g., control of lights, HVAC, music player). Figure 11 demonstrates a high-level view of one ATRACO system, which - as an example - implements the AS that supports the realization of goal *"Feel comfortable after work"*. Each member of the AE is assumed to have its own local ontology, which stores its state, properties, capabilities and services. Thus, all the data concerning the device and/or service is stored in its local ontology. Figure 11 shows the local ontologies in light green with several exemplary values which can be maintained by the corresponding entity.

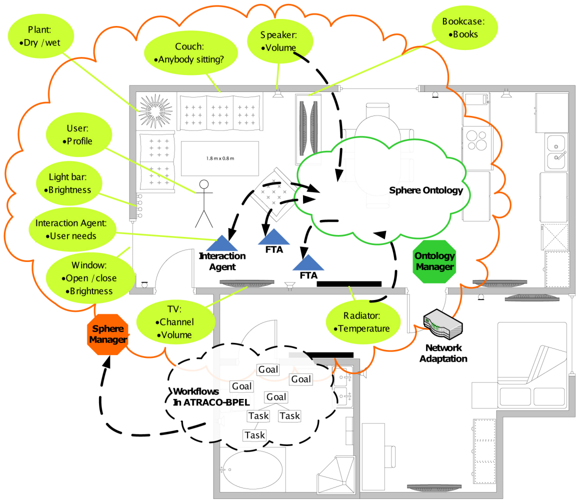


Figure 11: High-level view of one instance of an ATRACO system

In this example, the user's goal is to feel comfortable in his/her living room, no matter what the season is or the outside weather conditions are. The matching of "Feel comfortable" goal with available device local ontologies makes the Planning Agent (PA, see Figure 1) decide to instantiate the following tasks: adjustment of temperature, control of lights and selection of appropriate music. The AS goal is decomposed into a hierarchy of abstract tasks represented as a workflow which is specified in ATRACO-BPEL by the PA. The Sphere Manager (SM) forms or dissolves the AS for the specific user goal. The SM connects to device registry to get connectivity data (e.g., IP address, port) of the devices by the help of the Network Adaptation (NA) component which provides access to home peripherals and a mapping of all different network domains in the AIE to the IP level. Then, the SM initiates the various components (OM, IA and FTA) and performs dynamic service binding in order to execute the workflow. The number of FTAs initialized by the SM depends on the task workflow complexity. Afterwards, the SM calls the OM to create the SO. It is the responsibility of the SM to spawn the tasks as these are specified in the workflow and to satisfy them in the right order as indicated in the workflow.

A task may be assigned to FTA or IA. When the SM needs context state information, it queries the SO to get the required information. It then generates events that can be used by other components of the system. For example, the SM will continuously send light levels to the FTA so that it adapts the light in response. When an action is required to take place in the AIE, like starting the music player or switching on/off HVAC, the SM invokes the NA layer to change the state of the devices and the services. When a structural change in the AIE happens (e.g., a new device has joined in), the SM informs the OM and the latter starts a realigning process to update the SO with the new device. If a task fails for any reason (e.g., the device is switched off, is out of range), the SM attempts to find alternative parameters or services for that task. A request for semantic service discovery is initiated as an attempt to find a service replacement.

The iSpace at the University of Essex served us as a testbed in which we have deployed and tested the components. Whenever a user enters the iSpace, ATRACO automatically adapts lights and temperature to their preferences, as they are stored in their user profile. The user preferences might change at any moment, and for this reason, the SM creates an IA responsible for providing UIs adapted to the interaction context. In our scenario, the IA decides to instantiate a speech interface to allow the user to change the temperature at any time. This interaction is adapted to context changes. For instance, as the user goes near a screen, the IA pops up a graphical user interface (GUI) that allows the user to control the temperature and light levels. Such kind of interaction adaptation lets the user choose his/her favourite way of interacting. The FTA will learn from the changes done by the users as it modifies its user model to better match the user preferences by reproducing the same settings under similar conditions in the future. This allows the user to transparently specify or change his/her preferences by just altering the environmental settings. The role of NA is demonstrated with the transparent communication of devices and services residing in various different physical networks. For instance, X-10 lights can be controlled by the FTA based on the feedback given by a ZWave illumination detector. FTA is agnostic to the protocol used for the control of each individual device, as the devices are presented in a common way in NA.

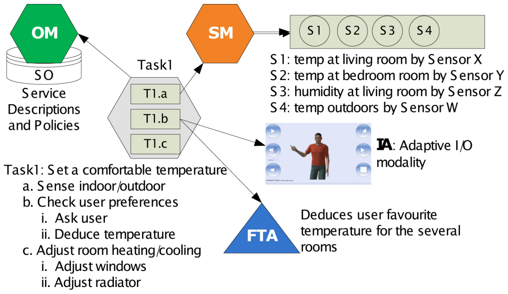


Figure 12: An example task supported by ATRACO

Figure 12 illustrates how the task "Comfortable temperature", which is a part of the user goal "Feel comfortable after work", is supported by the ATRACO system. The SM assembles the necessary services using specific sensor devices in order to execute the abstract task "Sense indoor/outdoor environmental conditions". The semantic mapping layer, represented by the OM is responsible for making semantic translations between the concepts perceived by the actors and the functionality provided by the devices. More specifically, it helps the system to deal with heterogeneity in resource descriptions, and it describes how the service discovery with the support of the ontology semantic descriptions should translate technical services or local resources (e.g., Sensor X, Y) into user perceived concepts (e.g., temperature, humidity). The FTA executes the subtask T1bii "Deduce favourite temperature". The role of the FTA is to support adaptation of the given subtask according to the user desires and behaviour and learn over-time in case the user overrides the automatically generated settings. The IA executes the subtask T1bi "Ask user for favourite temperature". The role of the IA is to support a multimodal front end to the user.

The starting point for running an AS is the generation of the corresponding workflow. Workflows are described in ATRACO-BPEL; however, they can be represented in a more user friendly way with activity diagrams. The diagram in Figure 13 illustrates the workflow for the "Feel Comfortable" AS of the example scenario. The diagram is annotated with labels from the source file in an attempt to close the gap between the high-level view of the diagram and the low-level view of the file. For example, the annotation in each box shows the activity type in the main sequence and the task name, the ontological searching term, as well as which ATRACO component, besides SM, is responsible for running specific parts of this task.

The technical requirements for the deployment and testing of the ATRACO system include: the runtime versions of the ATRACO components with the specified service interfaces; the devices serving the scenarios, wrapped as UPnP devices; the domain and resource ontologies; the workflows specifying the tasks in each AS; and various third-party run-time libraries. The deployment of the system has been done in two AIE testbeds using scenarios similar to the one discussed in this paper. The implementation technologies and tools used are based on open frameworks and are compatible with the SOA paradigm. Java is the main programming language, and UPnP enhanced with semantic descriptions is used as the communication middleware, overlaid upon TCP/IP, for the integration of iSpace devices and services. External Web Services is also possible to integrate through the NA component. The specification of the ontologies was performed using the Protg ontology development tool based on the OWL standard language. For ontology querying, the SPARQL language is used.

Although there are available (open source) execution engines for BPEL "programs" in ATRACO, we need to build a layer upon such engines as a proxy in order to process the parts of the workflow description that are ATRACO-specific. In addition, most engines do not allow dynamic binding and discovery of services. To address this limitation, the framework uses the SM as a proxy to communicate with service registries to obtain operational descriptions (e.g., UPnP or WSDL files) and instantiate services. This is achieved by encapsulating service search parameters in ATRACO-BPEL as inputs to the dynamic service binding process.

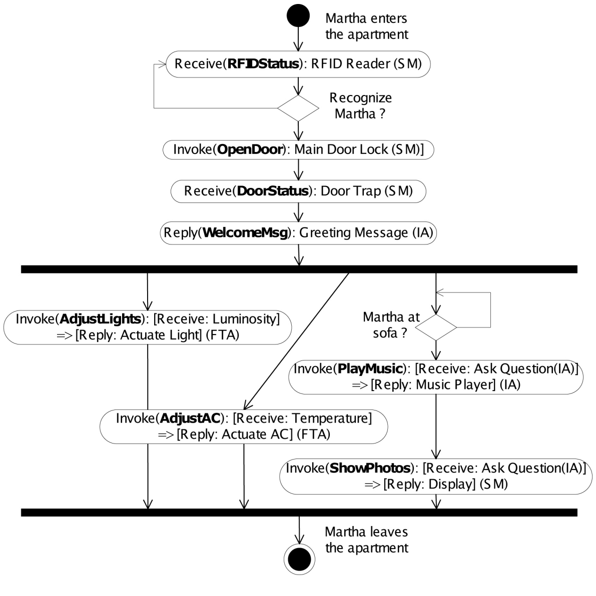


Figure 13: Annotated activity diagram for the "Feel Comfortable" AS

As a proof of concept, this prototype guided us during the whole design process. By raising the issues of realistic conditions - concurrency, resource limitations, inference delays, etc., it helped us distributing the roles of each component in a feasible implementation. The evaluation by users who actually played with the system has provided valuable feedbacks on our initial vision. As discussed in the following section, we will show that the system architecture was coherent by implementing it. Users felt that the system was indeed adapting to their needs and had a positive feeling about it. One of our concerns while designing this prototype was that people might have considered such adaptive features to be too intrusive. Yet, the study has shown that they rather liked it, as long as the system was truly fulfilling a need and not creating one.

We have also got some valuable feedbacks on how to drive the evolution of the system. By discussing with participants, we were able to discover new tracks for adaptation, mainly in the field of interaction. For instance, the idea about the GUIs that would move around and follow the users had come up just after the first version of the prototype. These sessions also helped us determine when and why the use of certain modalities like speech could be perceived as intrusive and when it was safe to use it. Finally, this prototype highlighted the technical issues that future designers of ambient systems will have to face: robustness of the network layers when the number of devices increase, need for computing power that allows ontological reasoning in various embedded devices, and consideration of delays in the information presentation.

10. EXPERIMENTS AND EVALUATIONS

From the outset, it was planned that the ATRACO prototype should be evaluated at two different stages during its development. The first evaluation was formative and conducted at a mid-point in the project life cycle. The results of this interim study informed the development about both the final prototype and the design of the final summative evaluation to be conducted at the end of the third year. For both studies, a working prototype was produced and installed at the iSpace in the University of Essex, which allowed the participants to interact with and experience the prototype in a relaxed and natural setting where they could imagine being their own home. As the vision of ATRACO is to present users multiple options and high levels of personalisation for managing their home environments, it was essential to shed light on their experience of interacting with the prototype, in other words, to understand how they feel about the system, its acceptability, its ease of use and the level of comfort or satisfaction it induces. Thus, both studies were conducted with a small number of participants using an in-depth qualitative interview approach after each had spent time interacting with the prototype.

The interim study involved nine participants (some screen shots of the evaluations using various participants is shown in Figure 14) who were asked to interact with the system in five short vignettes that demonstrated different aspects of the ATRACO vision. Each vignette prescribed either a single interaction or a flow of interactions between the participant and the system. The study has revealed that the majority of the participants were positive about the notion of living with some aspects of ATRACO in their home. However, key concerns were:

- Maintaining a sense of control over the technology (the system asked questions and demonstrated anticipation of user needs which some people were uncomfortable with).
- The variant nature of human moods in relation to an electronic presence (some days you might want ATRACO to anticipate your needs whereas other days you might not).
- The erosion of everyday life skills and domestic cultural norms and rituals.
- The risk of data loss and unauthorized access to the system.

The interim study has also concluded that in order to elicit a more distinctive response from the participants, the final study should allow them to be involved in more than one exposure to the prototype and to interact freely rather than their interactions being scripted in a vignette.

The final study involves eight participants (four males and four females) from a variety of ages and backgrounds as well as with different levels of enthusiasm for technology. The final prototype is robust enough to support unscripted interaction within each AS. Participants are therefore being asked to engage with the system in three different sessions and the associated ASs which are as follows:

- AS1 for 'Entertainment' based in the living room.
- AS2 for 'Working at home' based in the study room.
- AS3 for 'Sleep' based in the bedroom.

In each session, the participants are first given time to familiarise themselves with the prototype, and then, they are left to ‘play’ with the system and think about how the functionality relates to the patterns, routines and habits in their own lives. Each session is recorded in video and followed by an interview where the participant and the interviewer review the video together discussing the experience in detail. Gathering a variety of records regarding interaction with the ATRACO prototype revealed commonalities, themes, patterns and recursions that provided insight about the user’s experience of the prototype, suggesting the following:

—The perceptions/expectations of the prototype are strongly influenced by the devices and software interfaces that are currently used, such as remote controls, iTunes, iPod.

—With the use of multifunctional devices (iPad, laptop) users have found it difficult to distinguish between the controlling interfaces and the applications (an expectation that the photo frame should run on all devices in the AS)

—The users expect the different iterations of the ATRACO interface on different devices to be synchronised rather than working independently as they currently do.

—The ATRACO functionality in AS3 (’Sleep’ based in the bedroom) makes more sense and is more accepted by the participants than the same functionality in AS1 (’Entertainment’ based in the living room).

—The voice interaction is only acceptable if it works effortlessly - raised voices and curt commands are perceived to be unfriendly.

Hence, it was concluded that the ATRACO vision is socially acceptable and the various points raised above are going to be considered to enhance the ATRACO implementations.

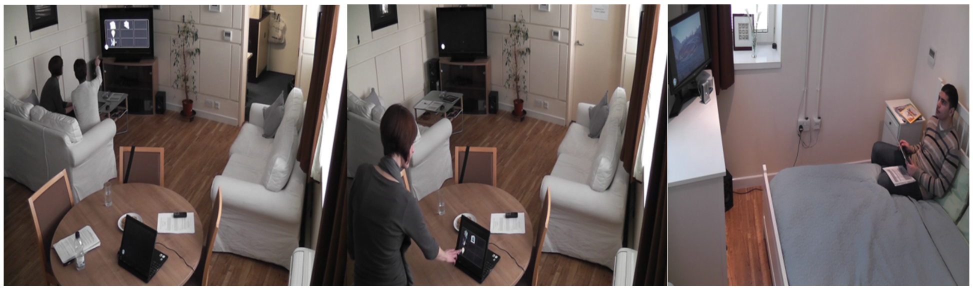


Figure 14: Evaluations and experiments conducted with various users

11. DISCUSSION AND CONCLUSIONS

In this paper we have introduced NGAIEs, which will provide the platform for the deployment of truly intelligent pervasive applications. Such applications will exhibit several systemic properties, like identity, autonomy, adaptation, and evolution in contrast to currently available applications, which are developed as compositions of services. To address the associated challenges we have applied an interdisciplinary effort from three different perspectives:

—*The conceptual perspective*, which focuses on concepts and models that capture the definition of NGAIEs at varying levels of abstraction.

—*The engineering perspective*, which focuses on the architectural challenges posed by the heterogeneous and dynamic nature of their synthesis.

—*The experience perspective*, which focuses on how people can accept the symbiosis within NGAIEs.

In this context, we presented a conceptual model of NGAIE-based applications, which adopts principles from object oriented architectures and autonomous systems. This model provides a user centric description of NGAIEs as platforms that host symbiotic ecologies of objects and services and support the deployment of pervasive applications as self-contained autonomous Activity Spheres (ASs).

Several research issues related to the deployment of NGAIEs are discussed in the paper and a layered architecture of NGAIEs (from top to bottom consisting of the layers: HCI, Semantic, Behaviour, Structure, and Communication) is proposed. Meanwhile, the paper focuses on the issue of heterogeneity in the descriptions of AS components and defines adaptation as the systemic mechanism that an AS develops in order to deal with this issue. It proposes the use of ontologies, both for the description of AIE resources and the implementation of adaptation. Next, the adaptation mechanisms that can be used to deal with different types of heterogeneity in each layer are described in detail (namely, Network Adaptation, Structural Adaptation, User Behaviour Adaptation, Semantic Adaptation, and User Interaction Adaptation).

As mentioned in Section 5, ATRACO applications are described by workflow specifications in the form of abstract services. Normally, workflow management systems have not been used for dynamic environments that require adaptive behaviour. On the contrary, in ATRACO, we depend upon adaptive workflows which need to react to varying environmental conditions. This transition from the static to dynamic and adaptive nature of workflows increases the runtime complexity of the management system, as the coordination mechanism must become more fault-tolerant. Our general idea then is that: since a workflow describes the relationship between services and if an agent is represented by such a service, then the relationship between the agents would be possible to specify. Following such a combined agent-based and SOA approach means that a workflow could be used to establish the initial relationships of the ATRACO agents. Then, an application can be specified first with a workflow description using ATRACO-BPEL that defines the most common scenario and fault conditions. Once the basic system has been deployed, the agents could be working proactively so that they can adapt to unforeseen circumstances and automatically handle the extension of the workflow description. For example, a FTA with learning capabilities and an IA, which are collaborating through the SM in a goal-related AS, can adapt heterogeneous artefacts within an activity sphere in order to support the user to fulfil tasks. This has been illustrated in the context of the 'Feel comfortable after work' AS described in Section 9.

On the other hand, a significant number of ontologies, specified in OWL, have been developed for the representation of ATRACO upper level concepts, devices and their services as well as attributes, users and their profile information. The ATRACO ULO serves as a knowledge base for the framework implementation, i.e., provides the necessary semantics to allow high-level exchanges, including brokering and coordination amongst software entities. A basic assumption of our approach is that these ontologies are heterogeneous as they are developed independently by device manufacturers, software developers, or non-domain experts. Establishing semantic links, that is, alignments, between these ontologies is a necessary precondition to achieve interoperability between agents, services, or applications using different individual ontologies. These alignments can then be used in order to translate requests and data between ontologies. For example, ATRACO scenarios generate questions such as which device offers a specific service, which devices are near the user, is it permissible or not for a user to carry out a specific activity, what are the preferences of a user according to a specific activity, etc. Therefore, when one looks for information, his/her search queries are likely to contain terms from various information sources that provide this kind of knowledge. To satisfy his/her information need, this scattered knowledge has to be gathered and integrated from disparate ontologies. The ontologies that address the aforementioned questions are the ATRACO domain ontologies, which gather the necessary knowledge about the devices and services, user profiles, etc. On the other hand, ontology alignment addresses the semantic integration of these ontologies by identifying semantically equivalent concepts in multiple heterogeneous ontologies. These concepts are then made compatible with each other through meaningful relationships. Hence, it is important to identify the correspondences between the concepts of different ATRACO domain ontologies. Finally, modelling the knowledge of the system in OWL-DL has resulted in some performance issues. We have found that computing multi-criteria numerical conditions required the assertion of extensive number of facts resulting in growing response times. Moving such computations from the knowledge base to an external Java module has improved query times.

The ATRACO approach is unique as it integrates multi-dimensional adaptation mechanisms which we believe are crucial for supporting realistically the users' tasks in large, dynamic and complex AIEs. These combined features are not present at the same time to a single architecture in any of the systems that we have referenced in the previous sections. Besides the overall architecture unique properties of the ATRACO system include:

—A decentralized workflow execution model that on the one hand does not suffer from the inflexibility of static workflows, and on the other hand can avoid the computationally expensive cost of frequent re-planning of composite services, because of the agent-based proactive behaviour.

—Novel UBA mechanisms based on human-centred concepts which have not been shown in the literature. The application of the user-centred models has been based directly on novel approaches for general type-2 Fuzzy systems based on z-Slices and in particular zSlices based agreement modelling as achieved in the FTA component.

—A significant increase in transparency by moving from device-centred models to human centred-models, in turn increasing interpretability, trust and thus facilitating user adoption.

—Support for multi-user adaptation which has so far largely been disregarded in AIE research.

—The dynamical composition of interaction units (e.g., devices, software modules, etc.) to create an adapted user interface coupled with spoken dialogue adaptation features.

—A user-led approach to the development of multi user behaviour adaptation, resulting directly from the tight coupling of AIE research with social evaluation as part of ATRACO.

A prototype that realizes the ATRACO architecture has been implemented, which exhibits, to a certain extent, all the aforementioned features and types of adaptation. The system has been deployed in the iSpace at the University of Essex and tested by real users, with promising results. The next steps of the proposed research include (a) the deployment of more complex scenarios that contain multiple users and ASs deployed in the same AIE, (b) the design of mechanisms that support conflict resolution within multiple user - multiple ASs scenarios, (c) the design of more efficient ontology alignment algorithms and (d) the analysis of system scalability factors.

12. ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community's 7th Framework Programme (FP7/2007-2013) under grant agreement n 216837 and from the Trans-Regional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG). We would like to thank Ms Aysenur Bilgin for her great help and contributions to this paper. We would like also to thank Dr. James Dooley and Dr. Joy VanHalvert for their valuable contributions to the paper.

REFERENCES

Amigo project. http://www.hitech-projects.com/euprojects/amigo/.

ATRACO - Adaptive and TRusted Ambient eCOlogies: FET project within European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 216837. www.atraco.org.

Chapter 30. Internet protocols, internetworking technology handbook. Tech. rep., Cisco Systems inc

Eperspace project. http://www.ist-eperspace.org.

Home2015 project. http://home2015.i2r.a-star.edu.sg.

Sense project. http://www.sense-ist.org.

TEAHA PROJECT. HTTP://WWW.TEAHA.ORG.

WEB SERVICES BUSINESS PROCESS EXECUTION LANGUAGE (WSBPEL).

NGN@HOME ETSI INITIATIVE. HTTP://PORTAL.ETSI.ORG/AT/ATNGNSUMMARY.ASP.

, , , , , AND 2010. In .

ABOWD, G., MYNATT, E., AND RODDEN, T. 2002. The human experience. IEEE pervasive computing, 48-57.

ALEKSOVSKI, Z., KLEIN, M., KATE, W. AND VAN HARMELEN, F. 2006. Matching unstructured vocabularies using a background ontology. Lecture Notes in Computer Science 4248, 182.

ALEKSOVSKI, Z., TEN KATE, W. AND VAN HARMELEN, F. 2006. Exploiting the structure of background knowledge used in ontology matching. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC).*

BAKER, C. R., MARKOVSKY, Y., GREUNEN, J. V. , RABAEY, J., WAWRZYNEK, J. AND WOLISZ, A. 2006. Zuma: A platform for smart-home environments. In *submission to IEE IE06..*

BECKER, C. , HANDTE, M., SCHIELE, G. AND ROTHERMEL, K. 2004. PCOM - a component system for pervasive computing", Proc. of the 2nd IEEE PerCom, IEEE CS, pp. 67-76.

BELLIK, Y., REBA, I. , MACHROUH, E. , BARZAJ, Y. , JACQUET, C. , PRUVOST, G. AND SANSONNET, J. 2009. Multimodal Interaction within Ambient Environments: an Exploratory Study. In *Human-computer Interaction, INTERACT'09: IFIP International Conference on Human-Computer Interaction; Uppsala, Sweden.*

BERTI, S. AND PATERN, F. 2004. Migratory multimodal interfaces in multidevice environments. In *Proc. International Conference on Multimodal Interfaces. ACM Publ., 92-99.*

BESLAY, L. AND HAKALA, H. 2007. Digital Territory: Bubbles. In *European Visions For The Knowledge Age, A Quest For New Horizons In The Information Society, Paul T Kidd (Ed.), Cheshire Henbury Publ.*

BIDOT, J., GOUMOPOULOS, C. AND CALEMIS, I. 2011. "Using AI Planning and Late Binding for Managing Service Workflows in Intelligent Environments", Proc. of the 9th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2011, IEEE CS.

BOUQUET, P. , EUZENAT, J. , FRANCONI, E. , SERAFINI, L., STAMOU, G. AND TESSARIS, S. 2004. Specification of a common framework for characterizing alignment. Tech. rep.

BRIAN, B. B., MEYERS, B., KRUMM, J., KERN, A. AND SHAFER, S. 2000. Easyliving: Technologies for intelligent environments. Springer-Verlag, 12-29.

BRUSILOVSKY, P.2001 Adaptive Hypermedia. User Modeling and User-Adapted Interaction 11, 87-110.

CALVARY, G., COUTAZ, J., THEVENIN, D., LIMBOURG, Q., BOUILLON, L. AND VANDERDONCKT, J. 2003. A unifying reference framework for multi-target user interfaces. Journal of Interacting With Computer Vol. 15/3,

COEN, M.1998 Design principles for intelligent environments. In *Proc. of the National Conference on Artificial Intelligence. JOHN WILEY and SONS LTD, 547-554.*

DAVIDSSON, P.AND BOMAN, M. 2005. Distributed monitoring and control of office buildings by embedded agents. In *Journal of Information Sciences* 171, 4, 293-307.

DE VICENTE, A. J., VELASCO, J. R., MARS-MAESTRE, I. AND PARICIO, A. 2006. A proposal for a hardware architecture for ubiquitous computing in smart home environments. In *Int. Conf. on Ubiquitous Computing: Applications, Technology and Social Issues.*

DEARLE, A., KIRBY, G., MORRISON, R., MCCARTHY, A., MULLEN, K., YANG, Y.,CONNOR, R.,WELEN, P. AND WILSON, A. 2003. Architectural support for global smart spaces. In *Lecture Notes in Computer Science,* 153-164.

DIETERICH, H., MALINOWSKI, U., KHME, T. AND SCHNEIDER-HUFSCHMIDT, M. 1993. State of the art in adaptive user interfaces. Human factors in information technology 10, 13-13.

DOCTOR, F., HAGRAS, H.AND CALLAGHAN, V 2005. An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans, Special Issue on Ambient Intelligence.

DUCATEL, K. , BOGDANOWICZ, M. , SCAPOLO, F. , LEIJTEN, J. AND BURGELMAN, J. C. 2001. Scenarios for ambient intelligence in 2010. In *Tech. rep., IST Advisory Group. February.*

EISENSTEIN, J., VANDERDONCKT, J. AND PUERTA, A. 2001. Applying Model-Based Techniques to the Development of UIs for Mobile Computers. In *Proc. of the 6th international conference on Intelligent User Interfaces. ACM Press Publ., Santa Fe, New Mexico, USA, 69-76.*

ERL, T.2005 Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR. In .

EUZENAT, J. 2004. An API for ontology alignment. In *3rd International Semantic Web Conference, Hiroshima, Japan, Lecture notes in computer science*, 698-712.

EUZENAT, J., PIERSON, J. AND RAMPARANY, F. 2008. Dynamic context management for pervasive applications. The Knowledge Engineering Review 23, 01, 21-49.

FLORINS, M., TREVISAN, D. and VANDERDONCKT, J. 2004. The Continuity Property in Mixed Reality and Multiplatform Systems: a Comparative Study. In *Proc. of CADUI'04. Madeira Island, 13-16.*

FRASINCAR, F. and HOUBEN, G.-J 2002. Hypermedia Presentation Adaptation on the Semantic Web. In *Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347. Malaga, Spain*, 133-142.

GOUMOPOULOS, C. and KAMEAS, A. 2009. "Ambient Ecologies in Smart Homes", The Computer Journal, Oxford University Press, Vol. 52, No. 8, pp. 922-937.

GU, T., PUNG, H. and ZHANG, D. 2005. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications 28, 1, 1-18.

HAGRAS, H.2007 Embedding computational intelligence in pervasive spaces. IEEE Pervasive Computing, 85-89.

HAGRAS, H., DOCTOR, F., CALLAGHAN, V. and LOPEZ, A. 2007. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. IEEE T. Fuzzy Systems 15, 1, 41-55.

HEINROTH, T., KAMEAS, A., WAGNER, C.and BELLIK, Y. 2009. A proposal for realizing adaptive ambient ecologies through an ontology-based multi-agent system. In *2009 IEEE International Symposium on Intelligent Agents.*

JAHNKE, J. H., D'ENTREMONT, M. and STIER, J. 2002. Facilitating the programming of the smart home. IEEE Wireless Communications 9, 6, 70-76.

KAGAL, L., KOROLEV, V., CHEN, H., JOSHI, A. and FININ, T. 2001. Centaurus: A framework for intelligent services in a mobile environment. In *Proc. of the International Workshop on Smart Appliances and Wearable Computing (IWSAWC). 195-201.*

KOBSA, A., KOENEMANN, J. and POHL, W. 2001. Personalised hypermedia presentation techniques for improving online customer relationships. The Knowledge Engineering Review 16, 2, 111-155.

KNINGS, B., WIEDERSHEIM, B. and WEBER, M. 2010. Privacy Management and Control in ATRACO. Chap in Ambient Intelligence, series: Lecture Notes in Computer Science, Springer Berlin, 51-60.

KUMAR, R., POLADIAN, V., GREENBERG, I., MESSER, A. and MILOJICIC, D. 2003. Selecting devices for aggregation. In *Proc. IEEE Workshop on Mobile Computing Services and Applications. IEEE CS Press, 150-159.*

LIN, R.-T., HSU, C.-S., CHUN, T. Y. and CHENG, S.-T. 2008. Osgi-based smart home architecture for heterogeneous network. In *Sensing Technology, 2008. ICST 2008. 3rd International Conference on. 527-532.*

MATSUURA, K., HARA, T., WATANABE, A. and NAKAJIMA, T. 2003. A new architecture for home computing. In *Proc. IEEE Workshop on Software Technologies for Future Embedded Systems. IEEE CS Press, 71-74.*

MENDEL, J.2000 Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice Hall.

MOZER, M.2005. Lessons from an adaptive home. Smart environments: technologies, protocols, and applications, 273-294.

PAGANELLI, L. and PATERN, F. 2002. Automatic Reconstruction of the Underlying Interaction Design of Web Applications. In *Proc. of SEKE 2002. Ischia, Italy.*

PERUMAL, T., RAMLI, A. R., LEONG, C. Y., MANSOR, S.and SAMSUDIN, K. 2008. Interoperability among heterogeneous systems in smart home environment. In *SITIS '08: Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems. IEEE Computer Society, Washington, DC, USA,* 177-186.

RAMPARANY, F., BOISSIER, O. and BROUCHOUD, H. 2003. Cooperating autonomous smart devices. In *Proc. Smart Objects Conference (sOc'2003). 182-185.*

RANGANATHAN, A., CHETAN, S., AL-MUHTADI, J., CAMPBELL, R. H. and MICKUNAS, M. D. 2005. "Olympus: A high-level programming model for pervasive computing environments," Proc. of the 3rd IEEE PerCom, IEEE CS, pp. 7-16.

REMAGNINO, P. and FORESTI, G. 2005. Ambient intelligence: A new multidisciplinary paradigm. IEEE Transactions on Systems, Man and Cybernetics, Part A 35, 1, 1-6.

RIVA, G., VATALARO, F., DAVIDE, F.and M. ALCAIZ (EDS) 2009. Ambient intelligence: from vision to reality IST Advisory Group (ISTAG).

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.

ROSE, B.2001. Home networks: A standards perspective. IEEE Communications Magazine, 78-85.

RUTISHAUSER, U., JOLLER, J. and DOUGLAS, R. 2005. Control and learning of ambience by an intelligent building. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 35, 1, 121-132.

SABOU, D., D'AQUIN, D. and MOTTA ET AL. 2006. Using the semantic web as background knowledge for ontology mapping. In *Proc. of the ISWC Workshop*. 1-12.

SAMAAN, K. and TARPIN-BERNARD, F. 2004. Task models and interaction models in a multiple user interfaces generation process. In *Proc. of TAMODIA 2004. Vol. vol. 86. ACM Press, Czech Republic,* 137-144.

SCAPIN, D. and BASTIEN, J. 1997. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behaviour and Information Technology 16, 4, 220-231.

SHERWIN, A.1999. Internet home offers a life of virtual luxury. The Times, 10.

SIRIN, E.and PARSIA, B. 2007. SPARQL-DL: SPARQL Query for OWL-DL. In *3rd OWL Experiences and Directions Workshop (OWLED'07).*

SOMMER, R.1969. Personal Space: The Behavioral Basis of Design.

STANCIULESCU, A., LIMBOURG, Q., VANDERDONCKT, J., MICHOTTE, B.and MONTERO, F. 2005. A transformational approach for multimodal web user interfaces based on UsiXML. In *Proc. of the 7th international conference on Multimodal interfaces. ACM New York, NY, USA*, 259-266.

STEPHANIDIS, C. and SAVIDIS, A. 2001. Universal Access in the Information Society: Methods, Tools, and Interaction Technologies. UAIS Journal 1, 1, 40-55.

TARPIN-BERNARD, F.2006. Interaction Homme-Machine Adaptative. Habilitation diriger des recherches (hdr), Universit Claude Bernard de Lyon.

THEVENIN, D.and COUTAZ, J. 2002. Adaptation des IHM: taxonomies et archi. logicielle. In *Proc. of the 14th French-speaking conference on Human-computer interaction. ACM New York, NY, USA,* 207-210.

VALTCHEV, D. and FRANKOV, I. 2002. Service gateway architecture for a smart home. IEEE Communications Magazine 40, 4, 126-132.

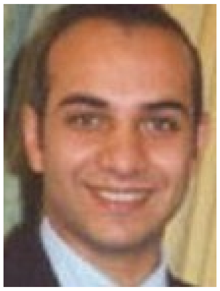
VANDERDONCKT, J., GROLAUX, D., VAN ROY, P., LIMBOURG, Q., MACQ, B. and MICHEL, B. 2005. A Design Space For Context-Sensitive User Interfaces. In *Proc. of IASSE 2005.*

VERMA, K., SIVASHANMUGAM, K., SHETH, A., PATIL, A., OUNDHAKAR, S.and MILLER, J. 2005. "METEOR-S WSDI: a scalable P2P infrastructure of registries for semantic publication and discovery of Web services," Information Technology and Management, vol. 6, no. 1, pp. 17-39.

WAGNER, C.and HAGRAS, H. 2008. zSlices-Towards bridging the gap between Interval and General Type-2 Fuzzy Logic. In Proc. of the IEEE International Conference of Fuzzy Systems, Hong Kong, China. 489-497.

YEN, J.1999. Fuzzy Logic-A Modern Perspective. IEEE Transactions on knowledge and data engineering 11, 1, 153-165.

Hani Hagras received the B.Sc. and M.Sc. degrees in electric engineering from Alexandria University, Alexandria, Egypt, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K. He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is also the Director of the Computational Intelligence Center and the Head of the Fuzzy Systems Research Group. His current research interests include ambient intelligence and computational intelligence, notably type-2 fuzzy systems, fuzzy logic, neural networks, genetic algorithms, and evolutionary computation. He has authored more than 250 papers in international journals, conferences, and books. He is an Associate Editor of the IEEE Transactions of Fuzzy Systems and on the Editorial boards of 5 International journals.



Christian Wagner is Transitional Fellow in Computer Science at the Horizon Digital Economy Research Institute and the School of Computer Science at the University of Nottingham, UK. He received his BSc in Computer Science, his MSc in Robotics and Embedded Systems and his PhD in Computer Science at the University of Essex in 2004, 2006 and 2009 respectively. His main research interests are centred around uncertainty handling and reasoning in uncertain environments through fuzzy logic and computational intelligence in general. He has published more than 30 papers and articles in international journals and at international conferences as well as two book chapters. He is organizer and chair respectively co-chair at the Special Session on Computational Intelligence for the Digital Economy and the Special Session on Type-2 Fuzzy Logic Theory at the World Congress on Computational Intelligence (WCCI) 2012. Over several years, he has held placements with several companies, including SES (Astra), Sun Microsystems and GE Fanuc Automation.

Achilles Kameas received his Engineering Diploma (in 1989) and his Ph.D. (in 1995, in Human-Computer Interaction), both from the Department of Computer Engineering and Informatics, Univ. of Patras, Greece. Since 2003, he is an Assistant Professor with the Hellenic Open University (HOU), where he teaches software design and engineering. Since 2007 he is Director of the HOU e-Comet Lab (Educational Content, Methodologies and Technologies Lab). He has participated as researcher / group leader or scientific coordinator in several EU R&D projects. Examples are the IST projects e-Gadgets, Astra, Plants, Social and Atraco and the LLP projects TIPS, VAB, PIN CompAAL and SONETOR. He has published over 100 journal articles, conference papers and book chapters, authored three university textbooks and co-edited more than five books and conference proceedings. His current research interests include engineering of ubiquitous computing applications, applications of ICT in education, semantic modeling and ontology matching.

Christos Goumopoulos received his diploma and Ph.D. degrees in Computer Science from University of Patras, Greece, in 1992 and 2000 respectively. He is currently a lecturer at Aegean University and a cooperating professor in the Hellenic Open University. Since 1992 he has been involved as member or leader of scientific and technology teams in several EU-funded R&D projects. His research interests include distributed computing, ubiquitous computing and awareness management, middleware, software engineering, resource scheduling and ontological knowledge representation. He has more than 45 publications to his credit in international journals, books and conferences in these areas.

Apostolos Meliones received a diploma with honours in Computer Engineering and Informatics from the Univ. of Patras, in 1994, and a PhD degree in distributed systems and telecoms from the National Technical Univ. of Athens, in 1999. From 1994 to 1999, he held a research position at NTUA. From 2001 until 2005, he was Technical Manager of Network Embedded Systems at Ellemedia Technologies, a Lucent product development partner. In Jan. 2006 he joined inAccess Networks where he coordinated the R&D Dept. He was the initiator, project manager or member of the management board of several EU and national research projects. He is currently a faculty member of the Dept. of Digital Systems at the Univ. of Pireaus. He is the author or co-author of more than 60 research papers in the areas of distributed computing, telecommunication systems, communication networks, fault-tolerant computing, middleware, embedded systems and pervasive computing.

Lambrini Seremeti received a degree in Mathematics (2000) and a Master's degree in Pure Mathematics (2004), both from the Department of Mathematics of the University of Patras, Greece. She also holds a degree in Applied Informatics in Management and Finance and is currently accomplishing her Ph.D dissertation in the Faculty of Sciences and Technology, at the Hellenic Open University. Her research interests are focused on ontology alignment and management. She has participated in the ATRACO project on the realization of trusted ambient ecologies. She has been a teaching and laboratory instructor at the Technological Educational Institute of Messolonghi, Greece, since 2004.

Tobias Heinroth received his diploma degree in computer science from the Ulm University, Germany, in 2007. He is currently pursuing the Ph. D. degree as a member of the Dialogue Systems Group at the Institute of Communications Engineering in Ulm. His current research focus lies on exploring ways of managing spoken dialogues within intelligent environments. His general research interests include Human-Computer Interaction, ambient intelligent environments, navigation systems and knowledge management.

Wolfgang Minker is a professor at the University of Ulm, Institute of Communications Engineering, Germany. He received his Ph.D. in engineering science from the University of Karlsruhe, Germany, in 1997 and his Ph.D. in computer science from the University of Paris-Sud, France, in 1998. He was a researcher at the Laboratoire d'Informatique pour la Mcanique et les Sciences de l'Ingnieur (LIMSI-CNRS), France, from 1993 to 1999 and a member of the scientific staff at DaimlerChrysler, Research and Technology, Germany, from 2000 to 2002.

Yacine Bellik is an assistant professor at LIMSI-CNRS (Laboratoire d'informatique pour la mcanique et les sciences de l'ingnieur). He is the head of the MIA (Modalities, Interactions and Ambient) research topic at LIMSI. "He holds a PhD and HDR in Computer Science. His research interests concern ambient intelligence and multimodal human-computer interaction.

Gaetan Pruvost is a PhD student at LIMSI-CNRS (Laboratoire d'informatique pour la mcanique et les sciences de l'ingnieur). He works in the MIA (Modalities, Interactions and Ambient) research topic at LIMSI with Yacine Bellik. He holds a master in Computer Science and his research interests concern ambient intelligence and multidevice human-computer interaction.