2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks

A Pervasive System Architecture that supports Adaptation using Agents and Ontologies

Achilles Kameas, Christos Goumopoulos
The Hellenic Open University and DAISy research unit
Patras, Hellas
{kameas, goumop}@cti.gr

Hani Hagras, Michael Gardner
University of Essex
Colchester CO4 3SQ, United Kingdom
{hani, mgardner}@essex.ac.uk

Tobias Heinroth, Wolfgang Minker
Ulm University
Ulm, Germany
{tobias.heinroth, wolfgang.minker}@uni-ulm.de

Apostolos Meliones, Dimitris Economou
inAccess Networks S.A
Athens, Greece
{meliones, decon}@inaccessnetworks.com

Yacine Bellik, Gaëtan Pruvost
LIMSI – CNRS
The National Center for Scientific Research
Orsay, France
{yacine.bellik, gaetan.pruvost}@limsi.fr

*Abstract***—In the context of the EU funded R&D project ATRACO we are developing a conceptual framework and a system architecture that will support the realization of adaptive and trusted ambient intelligent systems. Our approach is based on a number of well established engineering principles, such as the distribution of control and the separation of service interfaces from the service implementation, adopting a SOA model combined with intelligent agents and ontologies. Agents support adaptive task realization and enhanced human-machine interaction while ontologies provide knowledge representation, management of heterogeneity, semantically rich resource discovery and adaptation. ATRACO systems are dynamic compositions of distributed, loosely-coupled and highly cohesive components that operate in dynamic environments.**

Keywords-adaptation; ontologies; fuzzy logic; SOA; HCI

I. INTRODUCTION

Ambient Intelligence (AmI) is a paradigm that puts forward the criteria for the design of the next generation of ubiquitous computing environments [1]. In the AmI paradigm, intelligent computation will be invisibly embedded into our everyday environments through a pervasive transparent infrastructure (consisting of a multitude of sensors, actuators, processors and networks) which is capable of recognizing, responding and adapting to individuals in a seamless and unobtrusive way. Several approaches have been proposed for realizing Ambient Intelligent Environments (AIE), e.g., in [2], [3]. In these approaches, AIEs contain a dynamically changing set of "smart" objects or devices (hereafter called "artifacts"), which are able to perceive the environment, act upon it, process and store data, manage their local state, communicate and exchange data. As a result, artifacts may have physical properties (if they are physical objects), offer a set of services and can communicate with each other and the environment. In addition, the AIEs provide an infrastructure that supports services such as networking, communication, discovery, location and context estimation. These services are used by the artifacts that reside within this environment.

The next step is the design and development of totally adaptive ubiquitous computing systems, able to consistently operate in heterogeneous constantly changing AIEs. This Next Generation of AIEs (NGAIEs) will still be populated with numerous devices and have multiple occupants, but will inherently exhibit increasingly intelligent behaviour, provide optimized resource usage and support consistent functionality and human-centric operation.

In our attempt to realize adaptive AIEs, we have come across various research challenges, such as heterogeneity of artifacts, system transparency, discovery & management of various artifacts, and autonomous behaviour of learning agents. The ATRACO (Adaptive and TRusted Ambient eCOlogies) approach presented in this paper addresses these challenges and produces specifications and concrete realization of adaptable AIEs.

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 AIE. It is based on the notion of "bubble", which has been used to describe a temporary defined space that can be used to limit the information coming into and leaving the digital domain [4], which itself constitutes a "digitization" of the definition of personal space described as a "soap bubble" [5]. Inspired by 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 this data, leading to the specification of autonomous and coherent entities, which can adaptively execute on changing infrastructure.

Following a similar perspective, we consider artifacts as having an internal part that encapsulates their internal structure and functionality, and an external part that manifests their capabilities and can influence their environment.

978-0-7695-3908-9/09 $26.00 © 2009 IEEE
DOI 10.1109/I-SPAN.2009.81

Moreover, according to the paradigm proposed in [3], we consider them as basic building blocks of more powerful applications, which we model as AS; in principle, this approach can scale up smoothly by considering spheres to be more complex artifacts.

Previous research projects have applied the SOA paradigm to support dynamic service composition [6] or have combined SOA with ontologies to form a conceptual framework [7]. Most research efforts that have contributed to adaptation of ubiquitous applications during migration across different pervasive computing environments [8] provided little or no support for adaptation based on context information. Other research provided support for adaptation based on context information. In other research efforts, ontology techniques, such as merging and mapping have been adopted, but they all use ontologies as static objects. ATRACO architecture builds upon previous research by supporting multi-dimensional pervasive adaptation functionality into AIEs. Besides SOA a novel mechanism is proposed to achieve the different kinds of adaptation centered upon the management of knowledge, which is encoded in multi-layered ontologies, which are used by agents.

In the following section we will describe the proposed architecture focussing on all main components of the prototype. Before the paper finishes with conclusions a section describing the prototype itself provides details about the implementation.

II. ARCHITECTURE

The ATRACO approach uses a Service-Oriented Architecture (SOA) that enforces a clear distinction between service interfaces and implementation. SOA has been envisioned as an evolution of the component-based architectures centred on the concept of service [9] and thus is a convenient architectural style for realizing adaptable and reconfigurable systems. 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 on their own the adaptability and interoperability challenges of NGAIEs. 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 users adapt dynamically to the task the user wishes 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 Fig. 1). The ATRACO architecture consists of ontologies, active entities, passive entities, and the user who as the occupant of the NGAIE is at the centre of each AS. There are two main kinds of ontologies: local ontologies, which are provided by both active and passive entities and encode their state, properties, capabilities, and services and the Sphere Ontology (SO), which serves as the core of an AS by representing the combined knowledge of all entities (see Section II-II.B). Active entities are agents and managers. The agents (Fuzzy Task Agent and Interaction Agent) are responsible for automated adaptation, resolving conflicts, interacting with the user, and in general supporting the users achieving their goals. Two managers are responsible for the formation of the AS and for keeping the SO up-to-date (see Section II-II.A and II-II.B).

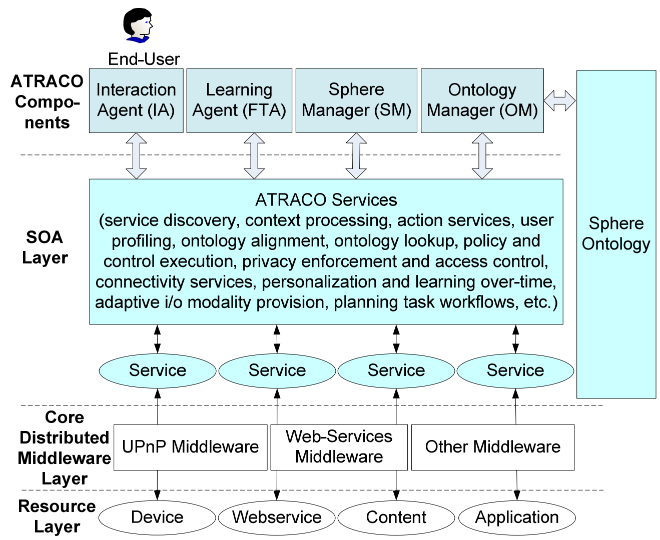

Fig. 1 ATRACO Architecture

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

Agents complement the SOA infrastructure by providing adaptation to user's tasks at an intelligent control layer higher than the SOA. Agents have a local knowledge base that contains domain independent rules describing their behaviour, which they adapt to each task, as they exchange semantically rich messages. In ATRACO, agents support adaptive task realization and enhanced human-system interaction. 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 to agents a common repository of system knowledge, policies and state. The ATRACO system takes users 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.

A. Sphere Manager

The Sphere Manager (SM) forms or dissolves an AS for a specific user goal. The SM is responsible for initializing the other system components (i.e., agents, Ontology Manager, etc.) and operates an event service for them. SM implements a semantic-based discovery mechanism (eRDP [10]) to bind the services in the concrete plan provided by the task model library onto actual executable services provided in the NGAIE. The SM composes an executable service workflow and implements an Execution Management and Control mechanism which is responsible for the execution of services on top of the SOA layer. Moreover, it supports the structural adaptation of AS providing for the persistent achievement of the goal when changes on the type of the available resources occur. To achieve this, it monitors the state of execution of the task workflow and might change the composition of services in case of any problem (i.e., a new device joining or a device going out of service).

The SM receives an event for starting the execution of an AS; such an event may originate from an agent, in response to an event, or from the user, for example using a User Interface (UI) provided by the Interaction Agent (IA). The SM connects to device registry to get connectivity data (e.g., IP address, port) of the devices. The SM connects to the network gateway (Connected Home Platform) 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 (Ontology Manager (OM), IA and Fuzzy Task Agent (FTA)) and performs dynamic service binding in order to execute the workflow.

The SM initializes one or more FTAs depending on the task workflow complexity. The FTA requires a set of devices and the user ID. The SM calls the OM to create the SO (see next section). The SM spawns the tasks as these are specified in the workflow and is responsible for satisfying them in the right order specified 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 FTA. When an action is required to take place in the AIE the SM invokes devices or services to change their state. 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 the task. A request for semantic service discovery is initiated as an attempt to find a service replacement.

The IA interacts with the user and supports the deployment of an AS to fulfil a user goal (stored in the user's profile) in case it requires an advanced service composition mechanism. Having a concrete workflow plan description, the Dynamic Service Binding component applies a semantic-based discovery mechanism and uses information about available services and context to discover suitable services or devices able to perform each task. The output of this process is an executable service workflow.

B. Ontology manager and ontologies

In general, an ontology is used as the means to share information among heterogeneous parties in a way that is commonly understood [11]. An ontology is a network of concepts and entities, which can be associated with different types of relations (the most common being the hierarchical association, or is-a relation). More concrete ontologies contain also instances of these entities (i.e., the bedroom TV as an instance of the entity TV) with specific properties and values (i.e., preferred channel in bedroom TV is channel 12). More powerful ontologies contain constraints and rules that cause inferences for the entities.

We assume that several (but not all) of the NGAIE components will contain their proprietary ontology, or set of meta-data, which describe properties, services, constraints and even state information of the component. Information about the user is contained in the user profile ontology, which contains instances such as personal data, location, preferences, and even goals and tasks. The NGAIE or the user profile may also contain policy ontologies, which constrain the interactions or the use of services.

As mentioned above, the deployment of an AS over an NGAIE requires the orchestration of available services and the inclusion of available resources. Because we expect that these will be heterogeneous and in order to ensure the user centric operation of the AS, we compose a SO. This encodes the information and knowledge necessary for sphere operation; it also provides context representation for the components of the AS. In general, the SO is formed by matching the local ontologies of the sphere resources, so as to ensure interoperability between the various services and devices. Moreover, the pertinent policy ontologies are matched to ensure correct sphere operation. Finally, the user profile ontology is matched to ensure that the sphere will serve a specific user goal (and its associated tasks) and take into account the user preferences and experience.

Ontology matching is the process of finding relationships or correspondences between entities of two different ontologies. Its output is a set of correspondences between two ontologies, that is, relations holding (with a degree of confidence) between entities of different ontologies. Current techniques for ontology matching require access to the internal structure of constituent ontologies, which is not acceptable in the ATRACO approach. That is why we choose to applying the ontology alignment technique. According to [12], the ontology alignment process is described as: given two ontologies, each describing a set of discrete entities (which can be classes, properties, rules, predicates, or even formulas), find the correspondences, e.g., equivalences or subsumptions, holding between these entities.

To the best of our knowledge, this is the first attempt that utilizes ontology alignment as an integrated mechanism to achieve context-based adaptation. A similar approach is proposed in [13], where an architecture that achieves adaptation based on context information is described, but is applied to a static pervasive systems architecture. The OM of ATRACO is responsible for managing the SO (as we stated above, local resource ontologies are managed by the resource entities in a way that is transparent to the ATRACO system). Thus, under the request of the SM, the OM produces ontology alignments, responds to queries regarding the state or properties of sphere resources, and creates inferences in order to enrich the SO.

The SO is a transitive, virtual component. It exists as long as the specific instantiation of the sphere is active. It contains the alignments of the local ontologies, which we choose not to merge in an integrated ontology, so as to achieve more efficient ontology management (i.e., when two or more ontologies are merged, it is not possible anymore to access the constituent ontologies, thus, it would not be possible to reflect in the SOs any changes caused in the local resource ontologies). In an NGAIE, there may exist resources that may be too simple to contain their own ontology (i.e., a lamp). The OM uses the ATRACO Upper Level Ontology (ULO), an ontology encoding the basic terms and concepts of the ATRACO world model, to merge all information it can find about them. When an AS is dissolved, the user profile manager stores the set of alignments in the user profile, thus ensuring that the experience gained can be reused in the future, if any of these resources participates in a new sphere or instantiation of the same sphere.

C. *Fuzzy Task Agent*

Inhabited AIEs face huge amount of uncertainties which can be categorized into environmental uncertainties and users' uncertainties. Uncertainties regarding the environment can be due to the change of environmental factors (such as the external light level, temperature, and time of day) over a long period of time due to seasonal variations. Furthermore the change of the sensors and actuators outputs due to the noise from various sources relates to theses uncertainties. In addition, the sensors and actuators can be affected by the conditions of observation (i.e., their characteristics can be changed by the environmental conditions such as wind, humidity, etc.). Changes in context, operation conditions, and wear and tear which can change sensor and actuator characteristics may also affect the environmental uncertainties.

The user uncertainties can be classified as intra-user and inter-user uncertainties. The former are exhibited when a user decision for the same problem varies over time and according to the user location and activity. The latter are exhibited when a group of users occupying the same space differ in their decisions in a particular situation. Thus it is crucial to employ adequate methods to handle the above uncertainties to enable the artifacts to produce the desired behaviour to perform a given task. In addition, there is a need to produce models of the users' particular behaviours that are transparent and that can be adapted over long time duration and thus enabling the control of the users' environments on their behalf.

Within the artifact adaption model, the process starts with the SO supplying the artifacts needed to perform a given task. The SO also provides the linguistic labels and the operational ranges of the variables involved with these artifacts. The AS adaptation level handles the fault tolerance issues with existing artifacts breaking down or new artifacts being introduced to the system by searching and exposing a suitable replacement artifact if available. Artifact adaptation deals with developing the strategies that will allow the artifacts to adapt to the uncertainties associated with the changes in the artifacts characteristics, context as well as changes in the user(s) preferences regarding these artifacts and their operation. Hence, the artifacts will adapt by adapting the operation values associated with the linguistic labels of the artifacts according to the changes in the artifact characteristics and context. In addition, the artifact will also adapt to the user(s) changes in desire and preferences for the fulfilment of a given task.

The user behaviour adaptation will enable a set of artifacts (heater, inside temperature sensor and outside temperature sensor, etc.) to work together to satisfy the human behaviour to perform a given task. Thus the FTA will oversee the realization of given tasks within a given AmI space based on the behaviours and desires of the user(s). These agents are able to learn the user behaviour and model it by monitoring the user actions. The agents then create fuzzy based linguistic models which could be evolved and adapted online in a life learning mode while handling the faced short and long term uncertainties. In this way, ATRACO would satisfy one of the main components of the AmI vision where the intelligence will not come from one information artifact, but emerges from collections of artifacts interacting and cooperating with each other, resulting in new behaviour and new functionality [14].

Recently, type-2 Fuzzy Logic Systems (FLSs), with the ability to model second order uncertainties, have shown a good capability of managing high levels of uncertainty. Type-2 FLSs have consistently provided an enhanced performance compared to their type-1 counterparts in real-world applications [15]. A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1] [16]. It has been shown that type-2 FLSs can outperform their type-1 counterparts [15]. However, no work has tried to approach the challenging area of developing AmI spaces that can handle the environmental uncertainties as well as the intra and inter user uncertainties in an environment.

Within ATRACO novel theoretical developments based on the zSlices theory [17] have been developed in order to minimise the computational costs of general type-2 FLS and to enable the application of general type-2 FLS is real world AIEs. For the user behaviour adaptation, the FTA will learn and adapt its rule base to face the changes in the user desires and preferences. Thus the agents will employ a modified architecture of the Incremental Adaptive Online Fuzzy Inference System (IAOFIS) technique reported in [15] while employing general type-2 fuzzy systems to handle all the faced uncertainties.

D. *Interaction Agent*

The IA (see Fig. 1) offers services to communicate with the user. One of the biggest problems in AIEs is how to achieve user's acceptance. Users might indeed find the system intrusive if it controls their environment on its own.

Consequently, we believe that it is necessary to inform the users of important changes, to answer their questions and to empower them with control over their environment. The IA is thus responsible for interfacing the system with the user. Of course, the IA can also be used by the system in order to get some input from the user, which leads to another interesting research issue: collaboration between the user and the system. This collaboration aspect can be used in order to support the user to select which tasks should be executed by the system and how they should be realized. In NGAIEs, the interaction context can change frequently (due to user's moves, environment changes, user's preferences). In order to provide smooth interaction, one of the most important roles the IA is to adapt the interaction to the context. Based on our experiments with the prototype, we have extracted several issues which are specific to this kind of adaptation. A main issue is the decentralization of the interface. The user interface (UI) links a user with an application kernel (core application) that can be remote. In NGAIEs, the UI needs to be completely separated from the core application and become available across a network. The UI rendering itself might also be distributed across different devices – for instance, a screen might present a UI for light regulation while this functionality is also accessible on speech modality via various microphones disseminated in the environment.

Another issue is that the user does not know how devices are interconnected. In NGAIEs, the user is immersed in a multitude of small devices wirelessly connected. Thus, the existence of a link between input and output devices is not physically visible. The UI has to maintain the user aware of these links to avoid user misunderstandings. Furthermore the availability of interaction devices can't be forecast. Inputs and outputs have to be adapted to available devices at a given moment in a given environment, which means that the UI should not rely on predefined input or output devices.

In order to solve such issues, we propose to use distributed multimodal interaction widgets [18]. This way, user's task can be distributed among available modalities and devices at runtime. Furthermore, we need to involve users in the composition of interactive devices. Among related work, IStuff [19] is an interesting framework proposing a platform-independent binding of devices for user interfaces. This binding can be done dynamically in order to enable a selection of input devices by the user. Hence, we propose to provide users with interaction techniques for the combination of different devices. However, since the system can learn the users habits and preferences (based on previous explicit interactions), it can then automatically try to make the best combination out of available devices and distribute the UI on it. This is a way to achieve adaptivity without giving the user the impression of overpowering him.

We consider speech to be the richest modality within the context of NGAIEs. It is obviously the most natural way to communicate while performing everyday's life activities. Three main classes of spoken dialogue, which mainly differ in complexity, can be recognized. The class with the lowest complexity consists of a set of main dialogues giving users the ability to control devices and services within ATRACO by the use of standard commands. More complex are proactive dialogues, which are initialized by ATRACO. The most complex class consists of so-called special purpose dialogues, which are generated depending on the context and give the user a chance to negotiate, ask for information or to provide input to the system. For the prototype described in Section III we have implemented a command-and-control dialogue that can be used instead of a graphical UI to control an MP3 Player. The IA adapts the interaction by automatically choosing the most appropriate UI.

E. *Connected Home Platform*

The Connected Home Platform (CHP) [20] is a technological solution we have used for prototyping a typical AIE, which seamlessly blends IP networking with a wealth of home automation functionality. The CHP provides uniform access to the controlled devices (including the full range of sensors and actuators) through an adaptation layer mapping all different network domains in the NGAIE to the IP level, and some basic services for task execution and event management. Within ATRACO, the CHP contributes to the realization of ASs under the orchestration of the SM. The CHP moderates access to NGAIE resources, collaborates with the OM by responding to queries regarding the state or properties of resources and propagating context changes, and maintains local device and policy ontologies.

The CHP integrates a set of Java tools and components that enable to quickly design, develop, and deploy services in an NGAIE utilizing the provided OSGi service platform and widely adopted automation technologies. The home controller is used to integrate connectivity with devices of various home control technologies, such as LON PL, LON TP, and Z-Wave (RF). The different network subsystems are interfaced in a common way through a device representation layer, known as ROCob, providing unified device representations. The main connected home functionality is actually offered through the device representation middleware, which can be accessed by applications.

III. 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). It was designed to make the user feel comfortable at home after work. The iSpace at the University of Essex served us as a test bed in which we have deployed and tested the components. Whenever a user enters the iSpace, the SM creates a new AS associated to the goal of making him feel at ease.

To this purpose, it automatically adapts lights and temperature to his preferences, as they are stored in his user profile. According to these preferences, the SM also decides to start playing music in order to provide a relaxing atmosphere. The user preferences might change at any moment, and for this reason the SM creates an IA responsible of providing UIs adapted to the interaction context. In our scenario, the IA decides to instantiate a speech interface so that the user can be asked for his preferred temperature.

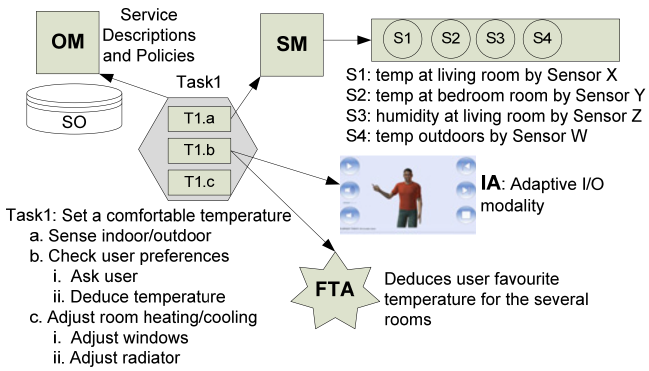

Fig. 2 An example task supported by ATRACO.

Fig. 2 illustrates how the task "Comfortable temperature", which is part of the user goal "Feel comfortable after work", is supported by ATRACO architecture. The SM for example 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 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) to user perceived concepts (e.g., temperature, humidity). The FTA executes the subtask T1bii "Deduce favourite temperature". The role of the FTA is to support for adaptation of the given subtask according to the user desires and behaviour and learn overtime 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 for a multimodal front end to the user.

IV. Conclusions

In this paper, we have presented the ATRACO approach towards realizing a new generation of adaptive AIEs. The ATRACO system aims to address the main challenges involved with realizing the AmI vision which are related to heterogeneity, transparency, discovery & management and most importantly adaptation. ATRACO supports five levels of adaptation: artifact adaptation, user behaviour adaptation (to learn and adapt to the changing user(s) preferences and environment), interaction adaptation (to provide transparency and ease of interaction to the user), network adaptation (to provide efficient discovery and management) and sphere adaptation (which handles behavioural and structural changes in the AIE where user tasks are realized). The paper has highlighted the various components of ATRACO and has presented the first integrated prototype of ATRACO, which has been successfully deployed in the iSpace. After evaluating the system with one and multiple inhabitants, we shall collect feedback on the usefulness of ATRACO concepts and requirements for the next system version.

Acknowledgment

The research leading to these results has received funding from the European Community's 7th Framework Programme (FP7/2007-2013) under grant agreement No. 216837 (project ATRACO) and from the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References

[1] P. Remagnino and G. L. Foresti, "Ambient Intelligence: A New Multidisciplinary Paradigm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 1, 2005.
[2] G. D. Abowd, E. D. Mynatt, and T. Rodden, "The human experience," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 48–57, 2002.
[3] A. Kameas, S. Bellis, I. Mavrommati, K. Delaney, M. Colley, and A. P. Cornish, "An architecture that treats everyday objects as communicating tangible components," in *Proc. PERCOM 2003*.
[4] L. Beslay and Y. Punie, "The virtual residence: Identity, privacy and security," *The IPTS Report, Special Issue on Identity and Privacy*, no. 67, pp. 17–23, 2002.
[5] R. Sommer, "Personal Space. The Behavioral Basis of Design." *eric.ed.gov*, 1969.
[6] AMIGO, "Ambient intelligence for the networked home environment project," EU Project, 2008.
[7] PalCom, "Making computing palpable," website: http://www.ist-palcom.org, 2008.
[8] A. Ranganathan, C. Shankar, and R. Campbell, "Application polymorphism for autonomic ubiquitous computing," in *Proc. MobiQuitous 2004*, pp. 402–411.
[9] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.
[10] C. Goumopoulos and A. Kameas, "Smart objects as components of ubicomp applications," *IJMUE*, vol. 4, no. 3, pp. 1–20, 2009.
[11] T. R. Gruber, "Towards Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies*, vol. 43, pp. 907–928, 1993.
[12] J. Euzenat, A. Mocan, and F. Scharffe, *Ontology Alignment: An ontology management perspective*. Springer, 2007, pp. 177–206.
[13] J. Euzenat, J. Pierson, and F. Ramparany, "Dynamic context management for pervasive applications," *The Knowledge Engineering Review*, vol. 23, no. 1, pp. 21–49, 2008.
[14] J. Wejchert, "The disappearing computer," Information Document, 1st Call for proposals, European Commission, Future and Emerging Technologies, 2000.
[15] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, "An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 41–55, 2007.
[16] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR, 2001.
[17] C. Wagner and H. Hagras, "zSlices: Towards Bridging the Gap between Interval and General Type-2 Fuzzy Logic," in *Proc. FUZZ-IEEE 2008*, pp. 489–497.
[18] M. Crease, "A toolkit of resource-sensitive, multimodal widgets," Ph.D. dissertation, University of Glasgow, 2001.
[19] R. Ballagas, M. Ringel, M. Stone, and J. Borchers, "iStuff: a physical user interface toolkit for ubiquitous computing environments," in *Proc. SIGCHI 2003*, pp. 537–544.
[20] N. Papadopoulos, A. Meliones, D. Economou, I. Karras, and I. Liverezas, "A connected home platform and development framework for smart home control applications," in *Proc. IEEE INDIN 2009*.