Ontology-driven composition of service-oriented ubiquitous computing applications

Eleni Christopoulou and Achilles Kameas

Research Academic Computer Technology Institute, Research Unit 3, Design of Ambient Intelligent Systems Group, 61 Riga Feraiou str. 26221, Patras, Greece
{hristope, kameas}@cti.gr

Abstract. The vision of Ambient Intelligence (AmI) implies a seamless environment of computing, advanced networking technology and specific interfaces. Technology becomes embedded in everyday objects and environments such as furniture, clothes, vehicles, roads and smart materials, and people are provided with the tools and the processes that are necessary in order to achieve relaxing interactions with this environment. The AmI environment can be considered to host several Ubiquitous Computing (UbiComp) applications; a UbiComp application can be considered as a result of the dynamic, ad-hoc composition of the services offered by the AmI environment and the objects therein. Key features of such applications are context-aware operation and emergent collective functionality. To achieve these, among others, one has to deal with heterogeneity and support adaptive composition and use. To do this, we propose to employ knowledge management and decision making techniques. As a first step, we present in this paper a service ontology and the management mechanisms we have developed in order to enable AmI artifacts to apply a common world model and a set of procedures that implement the composition of service-oriented UbiComp applications.

1 Introduction

The vision of Ambient Intelligence (AmI) implies a seamless environment of computing, advanced networking technology and specific interfaces [5] [8]. Technology becomes embedded in everyday objects and environments such as furniture, clothes, vehicles, roads and smart materials, and people are provided with the tools and the processes that are necessary in order to achieve relaxing interactions with this environment. The AmI environment can be considered to host several Ubiquitous Computing (UbiComp) applications; a UbiComp application can be considered as a result of the dynamic, ad-hoc composition of the services offered by the AmI environment and the objects therein.

Every new technology is manifested with objects that realize it; these objects may be new or improved versions of existing ones, which by using the new technology, allow people to carry out new tasks or old tasks in new and better ways. An important characteristic of AmI environments is the merging of physical and digital space (i.e. tangible objects and physical environments are acquiring a digital representation). The term “artifacts” is used for the objects in AmI environments that are augmented by adding to them sensing, computation and communication abilities.

The AmI artifacts differ from traditional objects in a number of properties and abilities. Specifically artifacts can communicate with other artifacts and can interact with the environment. Of special interest is the information that artifacts process, which can be descriptions of the context of use, data to be used for a task, guidelines on how to perform a task, messages to be sent or that have been received from other artifacts. The result of information processing is a set of services, that is, a set of abilities that appear in the digital space and relate to information.

Traditional objects have physical characteristics; mechanical ones also have capabilities, which describe the tasks they can do. The concept “affordance” describes the relationship between objects and the tasks that can be performed with them [7]. The artifacts possess two new affordances with respect to objects. The first one is the composeability; artifacts can be used as building blocks of larger and more complex systems. This is a consequence of them possessing a communication unit and requires universal descriptions of tasks and services. The second one is the changeability; artifacts that possess or have access to digital storage can change the digital services they offer. This means that the tangible object can be partially disassociated from the artifact’s digital services, as they are based on the manipulation of information. Both these affordances are result of the ability to produce descriptions of abilities, services and properties, which carry information about the artifact in the digital space. This ability improves object - service independence, as an artifact that acts as a service consumer may seek a service producer based on a service and not artifact description.

According to our approach the artifacts are treated as components of the UbiComp applications and offer a set of services. The composition of UbiComp applications can be based on the artifacts’ services. The target of this paper is to show that an ontology can accommodate the issues that emerge during the composition of service-oriented UbiComp applications and present the ontology that we developed for this reason.

The rest of the paper is organised as follows. Section 2 describes the composition of service-oriented UbiComp applications, the key issues that arise during this procedure and how an ontology can be used to accommodate them. In section 3 is presented the ontology that was developed accentuating on the representation of services offered by the artifacts. Section 4 through examples depicts the ontology-driven composition and deployment of service-oriented UbiComp applications. Section 5 targets to present how adaptive hypermedia techniques can be applied to UbiComp applications. In section 6 related approaches are presented. The paper closes with the conclusion and an outlook on future work in section 7.

2 Key issues in service-oriented UbiComp applications

In UbiComp environments artifacts have properties, like physical characteristics and sensors/actuators, which take values. These values determine an artifact’s state at a moment in time. The properties’ values can change because of an event; so events cause the change of an artifact’s state. In proportion to the state of an artifact a set of services is activated. The services that an artifact offers support the artifact’s usage, on the other hand the artifact' usages utilize the artifact's services. The services that an artifact can offer are determined by its physical and digital properties. A service can be seen as the publication of the state and properties' values of an artifact.

The Gadgetware Architectural Style (GAS) is a framework that supports the composition of UbiComp applications by treating the artifacts as components that offer services. The UbiComp applications are dynamic, distinguishable, functional and adaptive (re)configurations of associated artifacts, which communicate and/or collaborate in order to realize a collective behavior. Each artifact makes visible its properties, capabilities and services through specific interfaces (we'll sometimes use the term "Plugs"); an association between two compatible interfaces is called a "Synapse". The associations among the artifacts that users set depend on the services that artifacts offer and request and their compatibility. The (re)configuration of associations among the artifacts will enable users to set up their living spaces in a way that will serve them best. During the composition of UbiComp applications by associating the artifacts' services, a number of key issues that must be addressed arise. The descriptions of these issues follow.

2.1 Semantic interoperability

The composition of service-oriented UbiComp applications is based on the interaction and collaboration of both artifacts and services. A key issue of the artifacts and the services that they offer is their heterogeneity. So the challenge that we have to handle is the semantic interoperability among heterogeneous artifacts and heterogeneous services. In order to address the heterogeneity of artifacts we chose to base the interaction among artifacts on well-defined and commonly understood concepts, so as to enable a consistent and unambiguous communication. The common language that the artifacts have to use for their communication is represented by an ontology. The ontology that we developed is the GAS Ontology and its first goal was the description of the semantics of the basic terms of the UbiComp applications, such as eGadget (our term for artifact), Plug, Synapse, eGadgetWorld (our term for UbiComp application), and the definition of their interrelations. On the other hand in order to address the heterogeneity of services we first defined a semantic representation of the concept Service and then we designed a classification of a set of services based on common properties and characteristics.

2.2 Dynamic nature of UbiComp applications

One of the most important features of UbiComp applications is that they are created in a dynamic way. Users in order to create UbiComp applications have to select the necessary services and compose them consistently. Through plugs users perceive the services offered by the artifacts and can create and delete synapses between two compatible plugs. The compatibility of two plugs is determined by several factors e.g. the type of their input/output and the service that they offer/request, that must be represented into a formal form. Note that the synapses provide to users the necessary abstraction for service composition.

The dynamic nature of UbiComp applications depends also on artifacts mobility and failure that can cause the disestablishment of a synapse. In order to address artifacts' failures the UbiComp applications must be adaptive. A form of adaptivity is the automatic artifacts replacement. We selected to replace an artifact with another one that offers the same services. As this may be not feasible we decided to introduce the notion of "identicalness degree" between two artifacts. This degree depends on the services offered by two artifacts, their properties and their position into the proposed classification.

2.3 Context-awareness

An important issue of UbiComp applications is the context-awareness, as these applications must be able to perceive the current context and adapt their behavior to different situations. In UbiComp applications the term context is used to describe physical information, e.g. location and time, environmental information, e.g. weather and temperature, personal information, e.g. mood and activity. In our case, the term context refers to the physical properties of artifacts including their sensors/actuators and to their plugs that present services. Having described a service as the publication of an artifact's state and properties' values, the plugs provide context information. The user, by establishing synapses between plugs, both denotes his preferences and needs and defines the emerging behavior of the UbiComp application. Thus the UbiComp applications can demonstrate different behaviors even with the same context information depending on user preferences.

2.4 Adaptive services

In UbiComp applications the services provided to the users by the artifacts need to be adaptive to the changing requirements and needs of the users. Also they must be adaptive to changes of context information and fault tolerant. The possibility of adaptive service composition in a meaningful way is of special interest. The plug-synapse model that we use for the composition of service-oriented UbiComp applications supports service adaptivity. Initially this model captures users' needs, as users denote their preferences by establishing synapses. The plugs publicize context information, whereas the synapses represent artifacts' behavior dependent on context. One of the most important features of the plug-synapse model is that it provides users with an abstraction for service composition. Users have only to select the services that they want and combine them setting a synapse. The check for plugs compatibility secures the consistent service composition.

2.5 Semantic service discovery

The concept of service is fundamental, as the composition of UbiComp application depends on services. Users select the services that they want and form synapses seeking to achieve certain service configurations. Furthermore the services determine both artifacts' replaceability and plugs' compatibility. Thus the need for a service discovery mechanism is evident. A semantic service discovery mechanism is preferable in order to discovery the semantically similar services. This mechanism can be supported by the service classification that we designed and represented into an ontology.

3 An ontology for service-oriented UbiComp applications

The ontology that we developed in order to address the aforementioned issues in service-oriented UbiComp applications is the GAS Ontology [3] and is written in DAML+OIL. The basic goal of this ontology is to provide the necessary common language for the artifacts and services collaboration.

The artifacts' ontology contains the description of the basic concepts of UbiComp applications and their inter-relations; for the feasible collaboration of artifacts this knowledge must be common. On the other hand an artifact's ontology should both describe the way that the artifact is used and represent its acquired knowledge; this knowledge cannot be the same for all artifacts. So artifacts may end up having different ontologies. Since artifacts' collaboration is designed to be ontology-driven, the existence of different ontologies could result to inefficient interoperability. The solution that we propose allows each artifact to have a different ontology with the condition that all ontologies will be based on a common vocabulary. Specifically the GAS Ontology is divided into two layers: the GAS Core Ontology (GAS-CO); that contains the common vocabulary, and the GAS Higher Ontology (GAS-HO); that represents artifact's specific knowledge. Thus, all artifacts represent their different knowledge with common concepts.

3.1 The GAS Core Ontology (GAS-CO)

The GAS-CO represents the common language that artifacts use to communicate, so it must describe the semantics of the basic terms of UbiComp applications and define their inter-relations. It must also contain the service classification necessary for the service discovery mechanism. Note that it contains only the necessary information for the interoperability of artifacts in order to be very small and even artifacts with limited memory capacity may store it. The GAS-CO is static and it cannot be changed either from the manufacturer of an artifact or from a user. The graphical representation of the GAS-CO is on Figure 1.

The core term of GAS is the eGadget (eGt). In GAS-CO the eGt is represented as a class, which has a number of properties, like name etc. The notion of plug is represented in the GAS-CO as another class, which is divided into two disjoint subclasses; the TPlug and the SPlug. The TPlug describes the physical properties of the object that is used as an artifact like its shape; note that there is a cardinality restriction that an artifact must have exactly one TPlug. On the other hand an SPlug represents the artifact capabilities and services; artifacts have an arbitrary number of SPlugs. Another GAS-CO class is the synapse that represents a synapse among two plugs; a synapse may only appear among two SPlugs. Using the class of eGW the

GAS-CO can describe the UbiComp applications that are created by the users; an eGW is represented by the artifacts that contains and the synapses that compose it. The class of eGW has two cardinality constraints; an eGW must contain at least two artifacts and a synapse must exist between their SPlugs.

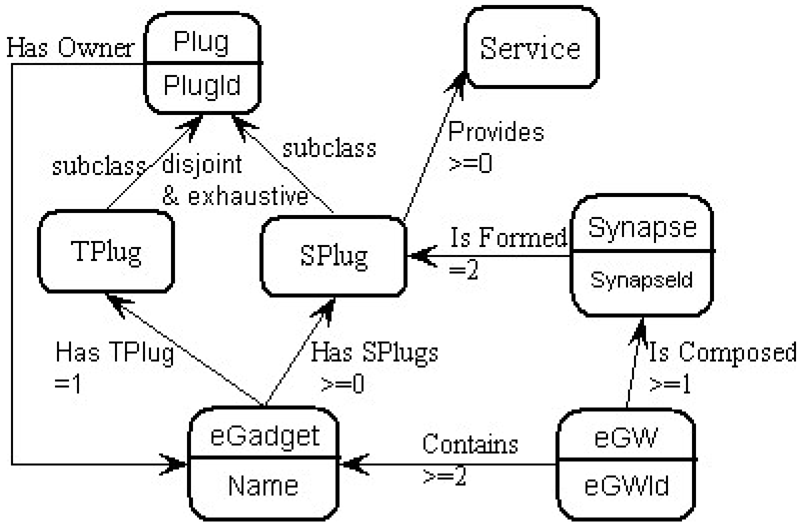

Fig. 1. A graphical representation of GAS-CO

3.2 The service classification

In order to define a service classification we first identified some services that various artifacts may offer; some results of this work are presented in Table 1. From these results it is clear that the services offered by artifacts depend on artifacts' physical characteristics and their sensors/actuators. The quality of services depends heavily on the placement of sensors/actuators at the artifact, e.g. if a weight sensor is placed on the left upper corner of an eCarpet and the user puts an eBook on the right down corner then the eCarpet will not perceive it.

Table 1. Services offered by artifacts

Artifact	Offered services
eLamp	switch on/off, light, heat
eBook	open/close, number of pages, current page
eDrawer	contains objects yes/no, number of objects, open/close, locked/unlocked
eMusicPlayer	sound, sound volume,, kind of music, play/pause/stop, next/previous track
eCarpet	object on it yes/no, objects' position, pressure, weight, frequency

Next we had to decide how we should classify the services. The classification proposals that we elaborated are the following: by object category, by human senses and based on the signals that artifacts' sensors/actuators can perceive/transmit. We decided to combine these proposals so that to describe a more complete classification.

So we initially defined the following elementary forms of signals that are used: sonic, optic, thermal, electromagnetic, gravity and kinetic. These concepts are divided into lower level services (subclasses); e.g. the sonic service may be music, speech, environmental sound, and noise. Additionally services may have a set of properties; e.g. sonic can have as properties the volume, the balance, the duration, the tone, etc.

Finally we enriched this classification by adding services relevant to environmental information, like humidity and temperature, and the concepts of time, position and movement.

3.3 The GAS Higher Ontology (GAS-HO)

The GAS-HO represents both the description of an artifact and its acquired knowledge; these descriptions follow the concepts defined in the GAS-CO. This means that the knowledge stored into the GAS-HO is represented as instances of the classes defined into the GAS-CO. Note that the GAS-HO is not a stand-alone ontology, as it does not contain the definition of the concepts that it uses, and its size doesn't need to be very small and depends only on artifact's memory capacity.

The information into GAS-HO is not static and it can be changed over time without causing problems to artifacts collaboration. As the GAS-HO contains both static information about the artifact and dynamic information emerged from its knowledge and use, we decided to divide it into the GAS-HO-static and the GAS-HO-volatile. The GAS-HO-static represents the description of an artifact containing information about artifact's plugs, the services that are provided through these plugs, its sensors and actuators, as well as its physical characteristics. On the other hand the GAS-HO-volatile contains information derived from the artifact's acquired knowledge and its use, such as the description of the synapses which the artifact's plugs are connected to and information about the services that other artifacts offer.

4 Ontology-driven composition of UbiComp applications

In this section we present an example of how we can use the GAS Ontology for the composition of service-oriented UbiComp applications. For the composition such applications artifacts must be GAS-compatible (use the GAS-Operating System [9]). The module that is responsible for the management of the GAS Ontology and supports the service discovery mechanism is the GAS Ontology manager [3]. The following example is based on the scenario, where a user creates its own "study" UbiComp application using two artifacts, an eBook and an eLamp.

The collaboration of these artifacts is feasible because both have stored the same GAS-CO. On the other hand the artifacts' GAS-HO ontologies are different. So eLamp's GAS-HO-static contains information about eLamp's plug "switch on/off" that offers a service type "light" and the eBook's GAS-HO-static contains the description of plug “open/close” that reflects the book’s state. The services that the artifacts offer can collaborate so these plugs are compatible allowing the user to establish a synapse between them. So when the user opens the eBook, the eLamp switches on, adjusting the light conditions to a specified luminosity level in order to satisfy the user’s profile. The knowledge emerged from this synapse is stored in both artifacts’ GAS-HO-volatiles. So the eBook “knows” that its plug “open/close” participates to a synapse with a plug that provides the service “light” with specific attributes e.g. luminosity.

If this synapse is broken e.g. because of a failure at the eLamp, a new artifact that offers a service type “light” must be found. The eBook’s GAS-OS in order to find such an artifact sends a message for service discovery to the other artifacts that participate to the same UbiComp application. This type of message is predefined and contains the type of the requested service and the service’s attributes. So an artifact may query just for a specific type of service or for a service with specific attributes. The GAS Ontology manager uses the service classification represented into the common GAS-CO of artifacts in order to find the artifacts that offer a similar semantically service with the one requested.

As the context information that is used in the UbiComp applications describes the physical and digital properties of artifacts, it is represented into both the GAS-CO and each artifact’s GAS-HO-static. The GAS-HO-volatile of artifacts contains mainly knowledge emerged from the synapses that compose an UbiComp application. So this information represents the artifacts’ behavior when they get context information through their synapses; these behaviors are defined by the user of the UbiComp application. As the GAS Ontology contains both context information and the description of the behaviors in proportion to context, provides the UbiComp applications with context-awareness.

5 Applying adaptive hypermedia techniques to UbiComp applications

Before presenting how adaptive hypermedia techniques can be applied to UbiComp application, we give an answer to the question: what can be adapted in UbiComp applications? Considering the UbiComp applications as dynamic (re)configurations of associated artifacts, few things can be adapted into these applications. First of all, the artifacts that take part in a UbiComp application can be adapted based on the user’s profile and his preferences. Additionally, the properties, capabilities and services that each artifact makes visible through specific interfaces can be adapted to the user’s needs and experience. The associations among artifacts can also be adapted based on the services that the artifacts offer and their similarity.

For the composition and deployment of an adaptive UbiComp application a critical issue is the user modeling; the representation and storage of the user profile. One significant difference between UbiComp and web-based applications is that the former are developed by the end-user. So at UbiComp applications it is not desirable to ask the user to provide to the system knowledge about its preferences, experience and goals. Though, this knowledge is necessary in order to provide an environment adaptive to users’ needs. In our approach users (dis)establish synapses between plugs and combine services in order to compose a UbiComp application. So a user denotes his preferences through the plug-synapse model. Also our system using a “fuzzy-logic agent” [6] tries to “learn” a user’s profile so that to adapt the environment to his needs.

In hypermedia adaptation content-level and link-level adaptation is distinguished as two different classes, the adaptive presentation and the adaptive navigation [1]. For the composition of UbiComp application we have taken into account some methods of content adaptation. In the AmI environment various artifacts exist that offer numerous services. When a user composes a UbiComp application, selects a set of artifacts and sets specific service combinations. So during the deployment of this application the user gets information relevant only to this application. Specifically since the user selects the artifacts that he wants to use, he can view the services’ and capabilities’ descriptions of only these artifacts. Additionally, our framework can support the presentation to the user of a text-based explanation of the artifacts’ usage and the functionality of their associations with other artifacts. Also, when users search for artifacts that offer a specific service, the system presents artifacts that offer semantically similar services.

The adaptive navigation in web-based systems attempts to guide the user through the system by customizing the link structure according to a user model. In UbiComp systems, the user needs an adaptive “mechanism” to guide him to compose a meaningful UbiComp application suitable to his profile. A kind of global guidance method that we use in our framework targets to inform the user about the artifacts that are available to him and their services as well as the service classification. A method of local guidance is the proposition to the user of semantically similar services that is supported by the service discovery mechanism and the service classification. The plug-synapse model provides a method of local orientation support; using this model our framework can show to the user the artifacts that he selected and the associations among them that he made.

6 Related work

Ontologies have been used in various infrastructures that support the composition of UbiComp systems. The UbiDev [10] is a homogeneous middleware that allows definition and coordination of services in interactive environment scenarios. In this middleware resource classification relies on a set of abstract concepts collected in an ontology and the meaning of these concepts is implicitly given by classifiers [14]. This approach is different than ours, because whereas they use an ontology for each application that includes several devices, our goal is to provide an ontology that drives the composition of various ad hoc UbiComp applications. Ontologies have also been used in the Smart Spaces framework GAIA [13] in order to address issues, such as the interoperability between different entities, the discovery and matching and the context-awareness [12]. The approach that the GAIA framework follows is fairly different to the one that we have proposed for the eGadgets project; an ontology server is used that maintains various ontologies. Another approach is the COBRA-

ONT [2], an ontology for context-aware pervasive computing environments. The Task Computing Environment [11] was implemented in order to support the task computing that fills the gap between what users really want to do and the capabilities of devices and/or services that might be available in their environments. This approach is fairly different to ours, since they use the OWL-S so that to describe the Web services and the services offered by the devices. An approach for applying adaptive hypermedia techniques to the composition of semantic web services is presented in [4]. Finally a very interesting work is the one made by the Semantic Web in UbiComp Special Interest Group [15]. The basic goal of this group is to define an ontology to support knowledge representation and communication interoperability in building pervasive computing applications. This project's goal is to construct a set of generic ontologies that allow developers to define vocabularies for their individual applications.

7 Conclusions and Future work

In this paper we presented how the composition of UbiComp applications can be based on artifacts' services, described the ontology that supports this composition and gave examples of deploying such applications. We described a service classification that assists a service discovery mechanism and presented how adaptive hypermedia techniques can be applied to UbiComp applications. The research presented in this paper has been carried out during the eGadgets project, a research project funded in the context of EU IST/FET proactive initiative "Disappearing Computer".

One of our imminent goals is to eliminate the limitation of the current version of the GAS Ontology that all artifacts have the same service classification, by adding to GAS Ontology manager the capability to map a service description to another one. Also we intent to use a set of methods in order to represent, acquire and refine the user model. Finally one of our targets is to develop the necessary mechanism in order to handle the existence of various users' profiles into the same UbiComp application.

References

1. Brusilovsky, P., 1996. "Methods and techniques of adaptive hypermedia", In P. Brusilovsky and J. Vassileva (eds.), Special Issue on Adaptive Hypertext and Hypermedia, User Modeling and User Adapted Interaction v. 6 n. 2-3, pp 87-129.
2. Chen, H., Finin, T., Joshi, A., 2004. An ontology for context aware pervasive computing environments, To appear, Knowledge Engineering Review - Special Issue on Ontologies for Distributed Systems, Cambridge University Press.
3. Christopoulou, E., Kameas, A., 2004. GAS Ontology: an ontology for collaboration among ubiquitous computing devices, To appear in Protégé special issue of the International Journal of Human – Computer Studies.
4. Conlan, O., Power, R., Barrett, K., Higel, S., O'Sullivan, D., 2003. "Next Generation Context Aware Adaptive Services", International Symposium on Information and Communication Technologies (ISICT 2003), Dublin (paper accepted).
5. Disappearing Computer initiative website http://www.disappearing-computer.net/

6. extrovert-Gadgets project website http://www.extrovert-gadgets.net
7. Gibson, J. J., 1979. "The Mifflin".
8. ISTAG website http://www.cordis.lu/ist/istag.htm
9. Kameas, A., Bellis, S., Mavrommati, I., Delaney, K., Colley, M., Pounds-Cornish, A., 2003. An Architecture that Treats Everyday Objects as Communicating Tangible Components. In Proceedings of the 1st IEEE International conference on Pervasive Computing and Communications (PerCom03). Forth Worth, USA
10. Maffioletti, S., Hirsbrunner, B., 2002. UbiDev: A Homogeneous Environment for Ubiquitous Interactive Devices. Short paper in Pervasive 2002 - International Conference on Pervasive Computing. pp. 28-40. Zurich, Switzerland.
11. Masuoka, R., Labrou, Y., Parsia, B., Sirin, E., Ontology-Enables Pervasive Computing Applications, IEEE Intelligent Systems, Sept/Oct 2003, pp 68-72.
12. Ranganathan, A., Campbell, R., 2003. A Middleware for Context-Aware Agents in Ubiquitous Computing Environments. In ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil.
13. Ranganathan, A., McGrath, R., Campbell, R., Mickunas, D., 2003. Ontologies in a Pervasive Computing Environment. Workshop on Ontologies in Distributed Systems at IJCAI, Acapulco, Mexico.
14. Schubiger, S., Maffioletti, S., Tafat-Bouzid, A., Hirsbrunner, B., 2000. Providing Service in a Changing Ubiquitous Computing Environment. Proceedings of the Workshop on Infrastructure for Smart Devices - How to Make Ubiquity an Actuality, HUC.
15. Semantic Web in UbiComp Special Interest Group. http://pervasive.semanticweb.org