An Ontology Based Approach to Interaction Ambient Design

Augusto Celentano, Andrea Okroglic, Fabio Pittarello
Università Ca' Foscari di Venezia, Dipartimento di Informatica
Via Torino 155, 30172 Mestre (VE), Italy
{auce,aokrogli,pitt}@dsi.unive.it

***Abstract*—We discuss the design of interaction ambients, which are ambients populated with fixed and mobile services, accessible through distributed interfaces. To make the interaction effective, activities and services should be designed as a function of the ambient, giving the user a coherent and clear context. A design approach is discussed, based on ontologies describing ambients, activities and services.**

I. INTRODUCTION

This paper discusses a methodology for designing interactive applications in complex physical environments, where human activity and interaction need to be adapted to the environment according to its functions. We shall denote the set of functions, roles and purpose of a physical environment with the expression *ambient semantics*; an environment in which services are offered through communication and information processing access points will be called *interaction ambient*.

In general terms, an application can be viewed as a set of cooperating services executing tasks and exchanging data, providing computation and information support, as well as specific results, to humans in a coordinated plan of actions. We use the term *service* in a generic sense, without subsuming any specific service architecture. Services execute (sub-)tasks, exchange information and objects, process information content and presentation, etc., according to a plan. Services may also be executed by humans: for example, an information service can be offered, in different contexts, by an automatic answering system, an interactive kiosk or a human guide.

The methodology represents a transition from context-awareness to *ambient-awareness*. Context-aware applications are able to adapt their behavior to a number of parameters that may belong to unrelated classes, such as location, time, device, user profile, etc., by selecting proper services among a set of alternatives, or by processing the information content to suit the constraints of the equipment. Context-aware applications are usually associated to mobile and ubiquitous devices: due to their limited processing and presentation features, adaptation helps the user to get information through these devices in less time and with less effort [1]–[4].

Shifting from context-awareness to ambient-awareness, adaptation depends not only on local properties of the user and her/his context, but also on the ambient *role*, i.e., on its meaning as a private or public place, having a defined function, a physical structure, and a space organization strongly depending on its function (such as a school, a hospital, an airport).

An ambient-aware application appears as a set of pervasive services distributed in a physical environment, accessible from local appliances, or within a wide area, or through mobile devices, delivering information and tangible objects, with which users interact in ways tailored to their context and to the properties, role and goal of the ambient.

Adaptation relates to the identification of the relevant properties of the environment navigated by the user. In most cases they are high-level properties (e.g., functional or social features) that can't automatically be inferred from measurements derived from a sensors' network. The combined geometric and semantic description of real environments is a promising research fields that can considerably help to improve the design of interactive distributed applications [5], [6].

II. ELEMENTS OF THE APPROACH

The study of the relations between the environment, its embedded services and the user activities evolving in it is a pre-requisite for optimizing the overall interactive system and obtaining the best results in terms of user satisfaction. We consider the different constraints that can affect the original state of the interactive system, discussing which are the potentialities and the optimization constraints in each situation.

A. Definitions

We use two basic definitions for the environment: *ambient* and *location*. The term *ambient* is referred to a spatial area organized in locations and populated by artifacts (e.g., urban infrastructures, buildings, embedded objects, etc.) and natural elements (e.g., rivers, plants, etc.); the term *location* identifies a partition in the space delimited by different types of objects (e.g., walls, floor, doors), morphologically meaningful for the user and suitable for the development of different classes of user activities. The definition for location is derived from a previous research activity targeted at identifying the structure of three dimensional environments belonging to the wide domain of mixed reality [7].

The term *interaction ambient* is referred to an environment characterized by the availability of embedded communication technologies that may be accessed by a visitor in order to get information and services. The term *service* denotes a work executed by a provider for a consumer, producing some kind

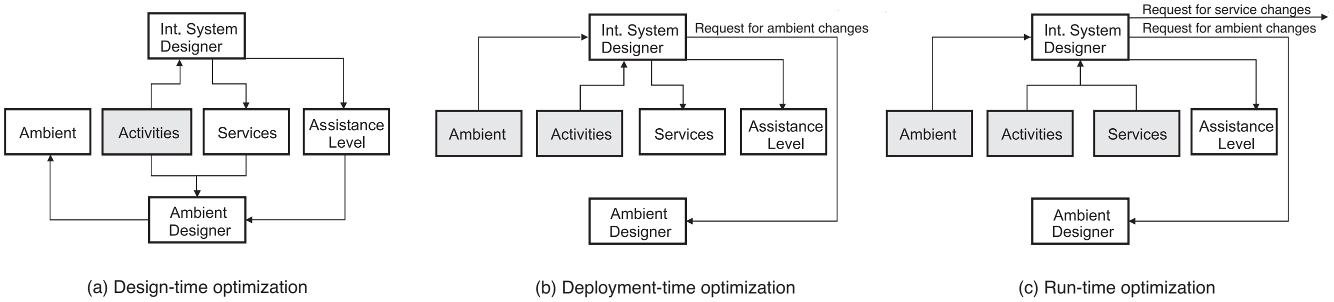
Int. System Designer
Ambient
Activities
Services
Assistance Level
Ambient Designer
(a) Design-time optimization
Request for ambient changes
(b) Deployment-time optimization
Request for service changes
Request for ambient changes
(c) Run-time optimization

Fig. 1. System Constraints and Optimizations.

of physical (e.g., a ticket) or abstract (e.g., a reservation) result. People moving through the locations can access services available in those places. The fruition of such services is regulated by a set of policies depending on the specific locations and on other system constraints, such as the physical structure of the ambient and the technological infrastructure.

According to the Activity Theory [8] and to Norman [9], we use the term *task* for referring to a situation with a single, well-specified goal (e.g., "perform a clinical test"). Users perform *activities*, that are groups of coordinated and integrated tasks. Activities are object-directed and can be therefore distinguished by the other activities in relation to the object itself. On the other side, tasks are made of *actions* that represent a more specific elementary level. In the context of this work the actions are supposed to happen in a single location. Actions may take advantage of *tools* (which include also service providers) that according to the Activity Theory are mediators between the subject and the object of the activity.

B. System Constraints and Design Optimization

We suggest a methodology to optimize the relations between the ambient, the services and the user actions, in order to obtain better results in the use of the whole interactive system. While in an ideal situation all the components of the system can be configured with a high degree of freedom, in practice the interaction designer often faces a number of constraints which make impossible to act on the components of the system as required.

In this section we consider three typical and real optimization cases, illustrating for each case the system constraints and their influence on the final design. Figure 1 illustrates the three different scenarios, showing for each situation the starting specification (in grey) and the outcome of the interaction designer activity. The design of the different components requires the intervention of a large pool of experts, but only the activity of the interaction designer and of the ambient designer are evidenced in this scheme.

Design-time optimization. This situation (Figure 1a) characterizes ambients (e.g., buildings) that have not already been built, and corresponds to the highest degree of freedom for the interaction designer who is able to obtain the best results in terms of interaction between users and services.

The initial requirement set is the specification of the activities that the different categories of users will have to perform in the final ambient. Such specification can be given in terms of functional diagrams defining in detail all the user activities and their relations. This scenario characterizes, for example, the situations where public administrators or private investors plan to set up a new building for satisfying the requests of the community or of a set of customers.

The interaction designer will base the specification of embedded information services for supporting the users' activity on the activity diagrams and on the desired level of user assistance for each service. The resulting compound document will be used by the ambient designer as a normative specification.

Deployment-time optimization. In this scenario the ambient is characterized by the presence of one or more ambients already built, where the interaction will happen; the services have not already been designed and deployed. The main constraint of this scenario (Figure 1b) is that the pre-existing physical ambient might discourage the implementation of certain services or specific user assistance levels because incompatible with the ambient features.

The interaction designer is able to identify the activities executed in the ambient, associating them to a semantic description of the ambient. The association allows the designer to discover and mark the points of weakness of the spatial configuration in relation to the user actions. The interaction designer will then elaborate the deployment of the information services according to the desired level of user assistance.

Run-time optimization. In this scenario the ambients and the services have already been designed and deployed. The design of the assistance to the user is the only contribution that can be given, since all the other components of the system, i.e., the ambient physical structure and the information services, are already defined and implemented (Figure 1c). In such situation the interaction designer will try to assure the best level of user assistance compatible with the constraints of the systems.

In case of impossibility to reach the planned level of user assistance there are two possible outcomes: a request for changing some services, to obtain an enhancement of the information infrastructure deployment; or a more radical request for changes in the physical ambient when service modification is not sufficient to grant the user assistance goals.

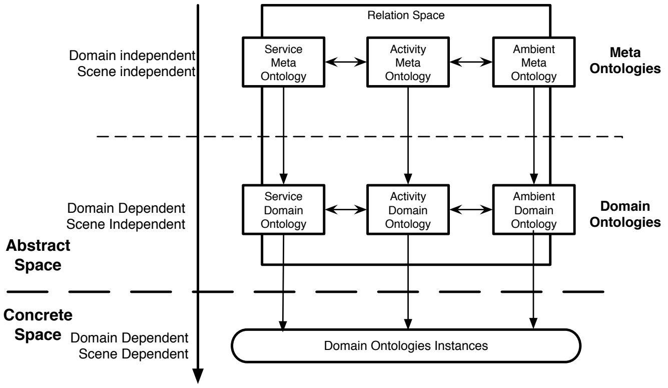

Fig. 2. The ontology and meta-ontology relations

III. Ontology Based Design

The design of the interactive environment is based on a set of ontologies describing the ambient, the application (in terms of services) and their relationships. Ontologies are introduced as a means to describe the environment components at different levels of detail, allowing both a precise description of the specific situation and different levels of generalization [10]–[12].

Three ontology types are defined: *ambient ontologies*, describing the physical environments and the semantic qualification of the objects contained, used to execute services and to interface the user with the application; *service ontologies*, describing the services available in the ambient and their results; *activity ontologies*, describing the relationships between services and ambients, necessary to execute in an effective way complex activities.

Ontologies for describing ambients and their properties have been discussed in previous work of one of the co-authors [13]. We shall focus the following discussion on service and activity ontologies.

The ontologies are defined at three levels (Figure 2): the upper level (*meta-ontology*) is used for expressing high-level concepts independent from the domain and the scene of the environment, related to the definitions of service, ambient and activity; the middle level (*domain ontology*) is used for declaring concepts and relationships domain-dependent but still independent from the specific physical environment; the lower level is characterized by the instantiation of the concepts described at the middle level, and describes a specific case.

The lower level gives a description of the instances of the different components of the interactive environment and of the type of interaction possible with specific scene constraints. It offers a description of the possibilities of a specific environment and of its services, that can be used by the components of a software architecture for controlling the delivery of information and interactive services.

The domain ontology assists in the definition of the lower level instantiations with a common representation language and can be reused in other situations belonging to the same domain. The existence of a common domain dependent ontological description helps to avoid incompatibilities at the instance level, supporting therefore the comparison and the merging of different scene-dependent and domain-dependent descriptions.

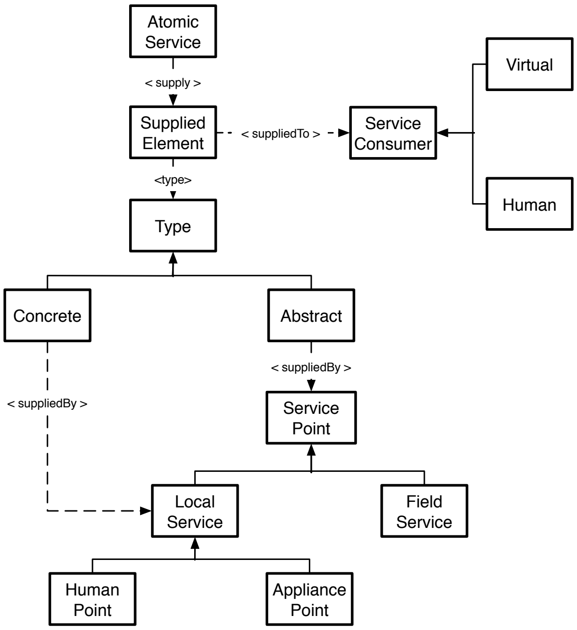

Fig. 3. The service meta-ontology

The upper level meta-ontology describes the relations among the concepts that are used to define ambients and services. Although far from the description of a specific domain, it represents a means for stimulating interoperability between different domain ontologies. Domain ontologies might describe concepts in very specific and incompatible ways; such descriptions may inhibit operations such as comparison and merging, which may be required when a system spans several domains.

Figure 2 illustrates also the relations with the physical ambient. The upper level contains the meta-ontologies that define, in general terms, the relations between the concepts that represent ambients, services and activities. It is domain independent because the concepts apply to any kind of service and activity. It is also independent of the physical ambient configuration, i.e., from the *scene*. Figures 3 and 4 illustrate the meta-ontologies for services and activities, respectively. The solid arcs denote sub-class relations, while the dashed arcs are semantic relations identified by evocative labels.

The *service meta-ontology* (Figure 3) defines the entities that conceptually describe a service. A service supplies an element to a consumer, that can be human or virtual (e.g., another service). The supplied element can be concrete or abstract, and is delivered at a service point. Abstract objects, such as information objects, are delivered in well defined locations (local services, e.g., a kiosk, a display, a desk with a human assistant) or can be accessed within an area (field services). Concrete elements can be supplied only locally.

The *activity meta-ontology* (Figure 4) defines an activity as

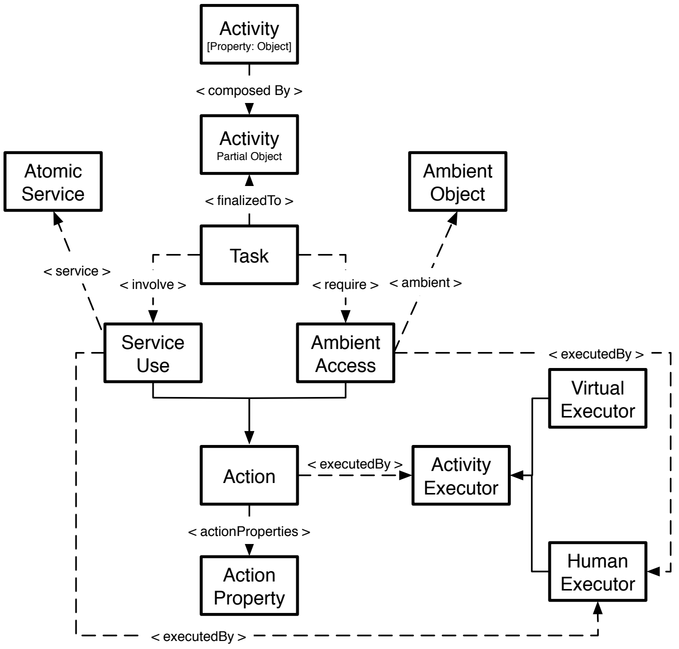

Fig. 4. The activity meta-ontology

composed by a set of tasks according to some plan [14], [15]. To execute a task, a user needs to use a service by accessing the ambient in which the service is supplied. Both service use and ambient access are elementary actions that can be executed through a virtual executor (e.g., a program) or a human executor (e.g., an employee at a desk).

The intermediate level of Figure 2 contains the domain ontologies that define what activities are executed in what types of ambients using what types of services. For example, they describe how a medical activity, say a medical test, is executed in a generic hospital, using services like *exam reservation*, *sample*, *medical record delivery*, and so on. It is obviously domain dependent, but it is still scene independent because the ambient is defined functionally and not physically. In Figure 5 the service domain ontology of a reservation for a medical test is illustrated.

The two upper levels of Figure 2 together refer to an abstract definition of spaces which does not take into account the exact morphology of the actual ambient.

The lower level of Figure 2 contains a set of instances of the domain ontologies for the concrete cases, referring:

- for the ambient, to the actual morphology of the space, in terms of areas devoted to user interaction;
- for the services, to the services which can be exploited in the ambient coherently with its characteristics;
- for the activities, to the association between the services and the points of supply: for local services, by specifying which parts of the ambients contain their appliances, for field services, which are the areas in which the service is provided (e.g., it can be accessed by a mobile phone, or by visual display, or by audio diffusion).

Further adaptation of the services can take place but is not

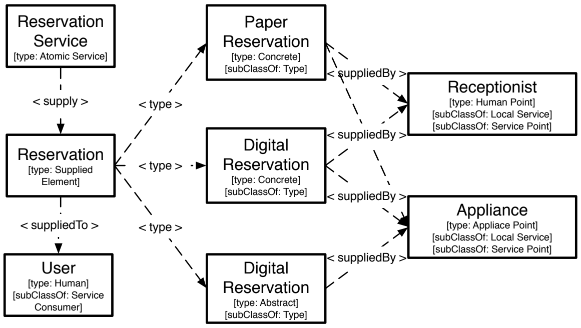

Fig. 5. A service domain ontology for a medical test reservation

described by the ontological model. It can be done according to the paradigm used for context-aware applications, based on sensing the user context and modifying accordingly the communication channels, the information presentation and the interaction.

In Figure 6 the relations among the ontology levels are depicted with reference to a service ontology describing a medical test reservation in a hospital. The service meta-ontology is instantiated, in the middle layer, into a set of tasks and suppliers which describe the operational structure of a reservation service for a medical test, according to the rules of the medical domain, as illustrated in Figure 5. The lower layer instantiates the tasks in a specific ambient[1] in which the reservation is made through an appliance (a ticket dispenser). A different instantiation could occur in an hospital enabled for on-line reservation, and the result would be a telematic reservation record delivered through a kiosk (or even through a wireless access point).

IV. Automatic ontology translation

The modelling of the different components that describe the spaces, the services and the use of the interactive environment has been checked by a tool for automating the mapping between the three different levels of semantic description: from domain and scene independency to a description dependent on the specific domain and on the actual scene. The tool has been implemented in Java using the toolkit associated to Protégé, a visual tool for modelling and exploring ontologies [16]. The prototype, on the basis of the semantic description of a meta-ontology written in OWL and to the specification of the instances of the classes for a specific domain, allows a designer to obtain the resulting OWL domain dependent ontology.

At the current development stage the tool proves the feasibility of introducing such automation, which can significantly simplify the design process, allowing the designer to focus on the definition of the relations and of their instances rather than on low level issues. The tool has been applied to the hospital domain, mapping the description of the service meta-ontology to a domain dependent medical test reservation service (as depicted in Figure 6). Further work will extend the tool to take

[1] The case has been evaluated with the Hospital of Cittadella, Italy.

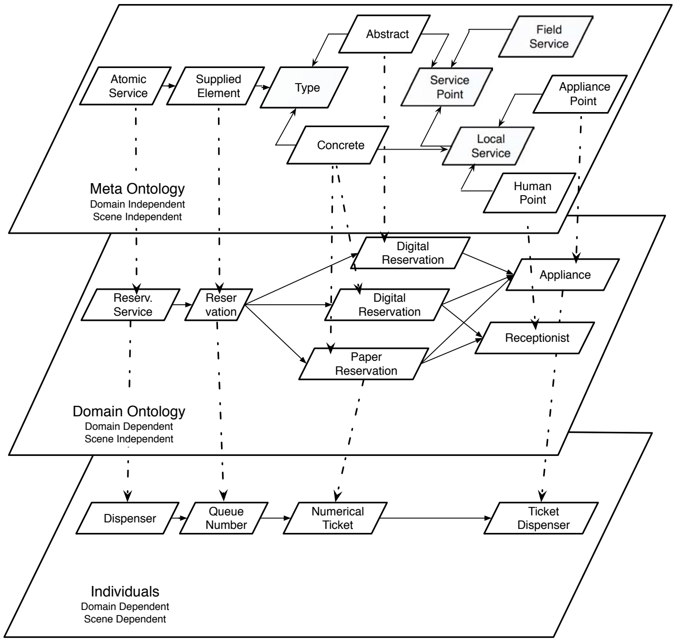

Fig. 6. Relations between ontology levels

care of the mappings among all the three ontological levels for each ontology type.

V. Conclusion

The use of ontologies for describing activities, services, ambients and their relations constitutes a step further towards the design of effective interactive systems in pervasive environments. As mobile computing spreads and the public administration regulations evolve, we can expect that a growing number of services are delivered to citizens through ICT, crossing the barrier of fixed workplaces in favor of distributed wireless access points. Ambients designed to deliver services on a human to-human basis need to be re-designed with a fresh attitude in mind, more abstract in considering the relations between the functions and the spaces in which they occur. But existing ambients will last for a long time, and adaptation as is approached today is no longer sufficient to satisfy the requirements of a really pervasive set of activities. The approach discussed in this paper could help to fill the gap between existing physical environments and new services delivery.

References

[1] G. Chen and D. Kotz, "A survey of context-aware mobile computing," Dartmouth College, Department of Computer Science, Tech. Rep. TR2000-381, 2000.

[2] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.

[3] A. Held, S. Buchholz, and A. Schill, "Modeling of context information for pervasive computing applications," in *Proc. SCI2002*, Orlando, FL, 2002.

[4] T. Lemlouma and N. Layaïda, "Context-aware adaptation for mobile devices," in *Proc. IEEE Int. Conf. on Mobile Data Management*, 2004, pp. 106–111.

[5] I. M. Bilasco, J. Gensel, M. Villanova-Oliver, and H. Martin, "An MPEG-7 framework enhancing the reuse of 3d models," in *Proc. Web3D 2006*, Columbia, ML, 2006, pp. 65–74.

[6] F. Pittarello and A. D. Faveri, "A semantic description of 3D environments: a proposal based on Web standards," in *Proc. Web3D 2006*, Columbia, Maryland, 2006, pp. 85–95.

[7] F. Pittarello, "Accessing information through multimodal 3d environments: Towards universal access," *Universal Access In the Information Society Journal*, vol. 2, no. 2, pp. 189–204, 2003.

[8] O. Bertelsen and S. Bodker, "Activity theory," in *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, J. Carroll, Ed. San Francisco: Morgan Kaufmann, 2003, pp. 291–324.

[9] A. Norman, "Human-centered design considered harmful," *Ambient intelligence: exploring our living environment*, vol. 12, no. 4, pp. 14–19, 2005.

[10] B. C. Vickery, "Ontologies," *Journal of Information Science*, vol. 23, no. 4, pp. 277–286, 1997.

[11] M. Uschold and M. King, "Towards a methodology for building ontologies," in *Procs. IJCAI-95, Workshop on Basic Ontological Issues in Knowledge Sharing*, Canada, 1995.

[12] D. Fensel, V. R. Benjamins, E. Motta, and B. J. Wielinga, "UPML: A framework for knowledge system reuse," in *IJCAI*, 1999, pp. 16–23.

[13] F. Pittarello and A. D. Faveri, "Improving access of elderly people to real environments: A semantic based approach," in *Proc. of AVI 2006, International Working Conference on Advanced Interfaces*, Venezia, Italy, 2006, pp. 364–368.

[14] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan, "Automated discovery, interaction and composition of semantic web services," *Journ. Web Sem.*, vol. 1, pp. 27–46, 2003.

[15] M. Vukovic and P. Robinson, "Adaptive, planning-based, web service composition for context awareness," in *Proc. Int. Conf. on Pervasive Computing*, Vienna, Austria, 2004.

[16] Stanford Medical Informatics, "Protégé programming developer kit," http://protege.stanford.edu.