

# Visual Design of Service Deployment in Complex Physical Environments

Augusto Celentano and Fabio Pittarello  
Dipartimento di Informatica, Università Ca' Foscari Venezia  
Via Torino 155, I-30172 Mestre (VE), Italia  
{auce,pitt}@dsi.unive.it

## ABSTRACT

In this paper we discuss the problem of deploying appliances for interactive services in complex physical environments using a knowledge based approach to define the relations between the environment and the services, and a visual interface to check the associated constraints, in order to design a solution satisfactory for the user.

## General Terms

Navigation, virtual and augmented reality, visual interaction, X3D

## 1. INTRODUCTION

Design of pervasive applications is an iterative process involving changes both in the application services and in the environment in which they are deployed; it is the result of a cooperation among a team of designers with different experiences and skills, examining the problem from different perspectives, and with different evaluation parameters in mind.

The team should include experts from many disciplines: architecture, information technology, HCI experts, etc., leading to a coordinated design of the physical environment and of the application. Coordinate design faces, in a unified process, the application services, their deployment in the environment, the user actions and the assistance to be provided for overcoming the difficulties due to user and environment limitations.

If a complex distributed application must be deployed in an existing physical environment, designers might not have sufficient degrees of freedom to optimally deploy the appliances for service access. The physical environment (a building, a set of rooms, an area) might discourage or prevent at all the implementation of some services and user assistance supports because incompatible with the environment structure. Nevertheless, the designers can collaborate to select a "good" deployment of interactive services within the constraints of the physical environment.

The interaction designer plays the most critical role in the team, identifying the activities executed in the environment, associating them to the environment locations and appliances and proposing to the environment designer and to the information engineer a suitable service deployment scheme compatible with the environment. The

association between activities and locations might lead the designer to discover points of weakness in the environment spatial configuration, which in critical cases could require the environment designer, i.e., the architect, to conceive a change in the environment physical structure to comply with the application to be deployed. Unfortunately, in most real cases, the change of the physical environment cannot be done and the application must be changed to fit the environment constraints, often lowering the quality of interaction.

The approach proposed in this work tries to limit such drawbacks, enabling the designers to propose alternative solutions that can be checked for consistency by the system.

## 2. INTERACTION AMBIENTS

Human activities take place in environments that we define *interaction ambients*, focusing on the availability of embedded interactive appliances and devices accessed by users in order to get information and services.

According to the *activity theory*, human activities are performed through the execution of *tasks* corresponding to specific user goals, according to some plan. Tasks are composed of *actions*, which are the basic level of operation on some device. Actions are executed by using *tools* which, in the information technology domain, include also service providers [2, 5].

Complex activities often require the user to move from place to place in the environment; hence, besides activity related tasks, also navigational tasks are important in interactive ambients. An interaction ambient is therefore best modeled as a set of connected locations, populated by objects and artefacts. Each location is defined by a physical or visual partition in the space (e.g., by walls or by pieces of furniture), morphologically meaningful for the user and suitable for the development of classified user activities. People moving through the locations can access services available in such locations by executing the task actions, which are therefore bound each to a single location.

From the information technology perspective, applications can be viewed as composed of *services*, works executed by a provider for a consumer; services produce both physical and abstract results, i.e., they may deliver concrete objects, such as a ticket, or produce changes in an information system, such as a reservation record. We distinguish between *field services* and *local services*: field services allow the user to access the service in a wide area, for example through a Wi-fi access point or through a large public display, while local services are bound to a device requiring user proximity or contact, for example a kiosk or an ATM. At design time, field services require the analysis of features such as range, covering and shields, while local services require the analysis of the physical placement.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

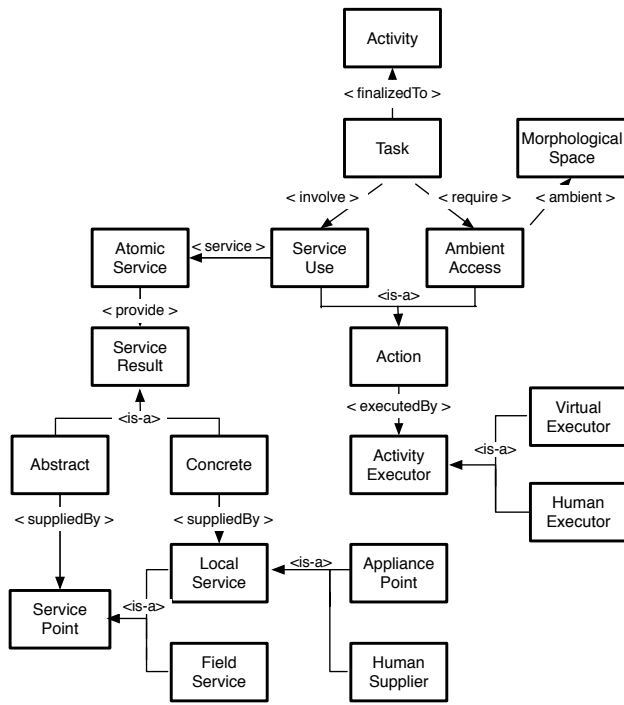


Figure 1: The activity and service ontology

### 3. PERVASIVE SERVICE DEPLOYMENT

In this section we present the functional architecture of a system supporting the designer in deploying pervasive services in complex interaction ambients. The system operates on a semantic description of the ambient, of the activities performed in it and of the available services, provided by a set of ontologies. An *ambient ontology* describes the physical environment and the objects in it, which are the interface between the user and the application services. A *service ontology* describes the services available and the concrete or abstract results delivered. An *activity ontology* relates services to ambients, describing how to execute in an effective way complex activities. Ontologies are defined at three levels: a meta-level describes the general structure of an ambient, service and activity, independent from the specific domain; a domain dependent level describes ambient, activity and service types relevant for an application domain; an instance level related to a case describes a specific physical environment, a defined set of activities and the set of services provided by that case.

Figure 1 shows a simplified view of the relations between the service and the activity meta-ontology components, which refer to the activity theory concepts, as illustrated in Section 2. An activity is composed by a set of tasks; to execute a task, a user needs to use services by accessing (hence navigating) the ambient in which they are available. Such elementary actions can be executed through a virtual executor (e.g., a program) or a human executor (e.g., an employee at a desk or the user him/herself).

Atomic services deliver concrete (objects) and abstract (information) results through field and local services. Local services can be supplied by appliance points or by human suppliers. Due to space limits, the reader is referred to [3] for details.

Figure 2 illustrates the functional architecture of a system able to support the interaction designer to deploy services in a way consistent with the environment constraints. The designer uses a visual interface based on a virtual reality paradigm for displaying a navi-

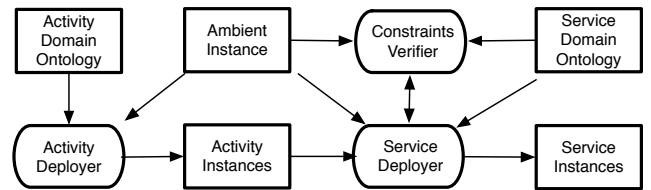


Figure 2: A functional architecture for service deployment

gable 3D model of the physical environment. The interface displays also the list of activities and tasks, allowing the designer to trigger the system components devoted to deploying the service appliances in selected locations.

The *activity deployer* takes as input the description of the activities at the domain level provided by the activity domain ontology, processes them together with the description of the physical environment, and produces as output the description of the activities deployed in that specific ambient. The deployment is based on the identification of the relevant functions of the environment locations and of the objects contained; the designer can examine the system proposal and modify it, e.g., by suggesting alternative locations or different paths to follow to execute the activity.

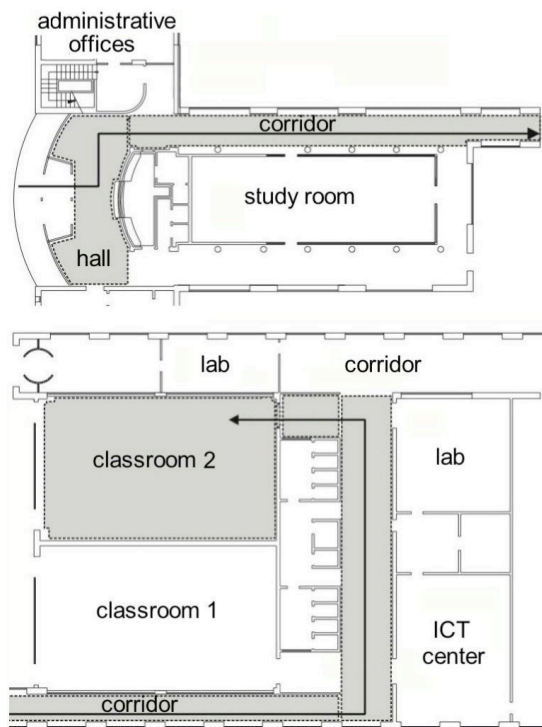
The *service deployer* places local service appliances and field service access points, defined by the service domain ontology, in selected locations, according to the designer choices. The technical features of the services concerning placement, access, use, safety, efficiency, etc., are part of the service ontology, and are expressed as constraints. Some constraints must be obeyed by all the services of a given type: for example, all the appliances for local services must be visible and reachable by users, and must have enough space around them to avoid obstructions. Other constraints are referred to specific appliances: for example, a telematic kiosk must be placed near to electric and network connections, next to a light source and in a safe place.

Table 1 shows a sample of constraints: they correspond to specific needs stemming from user, communication, safety and wiring requirements. User requirements are related to the ability to identify the presence of a service, to physically access the appliance if it is a local service, and to the ease of use. Communication requirements are related to the availability of a field service inside a given area, and are focused on avoiding discontinuities due to bad placement and occluding elements.

Table 1: Sample constraints for service appliances

appliance	req.	type*	description
kiosk	user	m	place next to the user's path
kiosk	user	p	place next to the user's path start
kiosk	user	m	place in the user's sight
kiosk	user	p	don't place in narrow connection spaces
kiosk	user	p	place against an opaque surface
kiosk	user	m	place next to a light source
kiosk	wiring	m	minimize distance from existing wiring
kiosk	wiring	p	place next to existing partitions
kiosk	safety	p	place against a robust surface
kiosk	safety	m	place in controlled locations
wi-fi	comm.	m	minimize occlusion
wi-fi	comm.	m	place on the ceiling or on top of vertical partitions
wi-fi	wiring	m	minimize distance from existing wiring
wi-fi	wiring	p	place next to existing partitions
wi-fi	safety	m	place in controlled locations

\* m=mandatory, p=preferred



**Figure 3: A map of two Computer Science Department areas**

Safety requirements are related to the safety both of the appliance and of the user, preserving them from intentional or accidental physical damage. Wiring requirements are related to the infrastructure needed to connect the service appliance to the power supply and to the communication network.

An important component of the architecture is the *constraint verifier*, that allows the designer to check the compatibility of the local and field services deployed with the constraints defined in the service domain ontology, with respect to the ambient instance features.

#### 4. A CASE STUDY

The building hosting the Department of Computer Science of Ca' Foscari University is divided into three areas connected by corridors and halls: an administrative area hosts the reception and administrative offices; an educational area hosts classrooms, undergraduate laboratories and the ICT centre; a research area hosts the teachers' and researchers' offices, the library and the graduate and post-doc labs. Study rooms and other services are distributed in the three areas, which therefore lack a strong separation of functions. Figure 3 shows a map of the first two areas.

Different classes of users populate the building for executing different types of activities. Because of the lack of a clear separation of functions in the building, casual visitors unfamiliar with the department physical structure may experience difficulties in finding the right place for performing their activity. While daily users need to be supported mainly for their core activity (such as attending lessons, consulting the library, etc.), occasional users, need assistance also for moving inside the building to locate the relevant rooms and services; navigation in the building becomes therefore an important task.

To exemplify the service deployment process, we consider an occasional user who has to attend a half-day seminar in classroom 2. Figure 3 shows the navigational path that the user should follow

to attend the seminar, superimposed on the department map. While being only a part of the user activity, the task of going to the right place requires more support than the task of attending the seminar because of the user unfamiliarity with the building structure.

The designer effort is thus focused on placing services for helping the user to identify the correct path leading to classroom 2. The designer chooses to deploy a display with an interactive map of the building, showing the services, the current activities and their location, next to the building entrance, which is the beginning of any path for users' activities. Such local service will give assistance to the casual users even when the front desk personnel, usually working at the reception next to the entrance, is temporarily unavailable.

As a local service, the display placement must obey a number of constraints, similar to the ones listed in Table 1 for kiosks. The constraint verifier can be used to check the features of the *interactive display* appliance and of the location selected by the designer, showing the relevant constraints. Constraints that are violated by the placement, such as placement in a narrow passage, or in an unsafe location, or occluded from the entrance door, can be detected and corrected.

#### 5. A VISUAL DEPLOYMENT INTERFACE

Figure 4 shows the proposal for the interactive deployment system interface. The upper menu shows the main functions that enable the user to choose the scenario to work with (item *Model Selection*), to deploy the activities (item *Deploy Activities*), to deploy the services (item *Deploy Services*) and to trigger the deployment checking system (item *Check Deployment*). The last item of the menu (item *Generate Docs*) generates the technical documentation produced by the system as part of the service deployment solution.

The central part of the screen is devoted to the 3D model of the physical environment. The ambient description is based on a proposal, described in [4], that takes into account not only the geometry of the objects represented, but also their high-level semantics: for example, an information panel is described both in terms of geometric primitives and with labels identifying its function. The approach is based on web standards: semantic web layers are used for defining the classes of semantic objects and their relations. The X3D standard [1] is used for describing the geometry of the 3D environment and its association with the higher-level semantic objects. While the semantic web layers define a scene-independent ambient ontology, the geometric description qualifies the specific ambient instance.

During the initial interaction phase, the interaction designer selects the X3D model to work with, which is visualized in the 3D window; Then, he/she defines the activities associated to the environment (see Figure 4.a). The activity deployer component supports the designer proposing, based on the knowledge of the environment semantics, a navigational path leading the user from some initial position of the activity (e.g., the building entrance) to the place where the activity is done or completed. In literature several navigation algorithms are discussed, which can be applied for accomplishing this task, that will not be detailed here. The interaction designer can modify the single steps of the path suggested by the system, using the menu on the right of the screen. Alternatively, he/she can design the user path from scratch, navigating the 3D environment and defining, as single steps, a set of views corresponding to subsequent current positions on the scene. The interaction designer can also choose to add new activities using the menu on the left side of the screen. At the end of the process, a set of paths associated to different activities, corresponding to the user movement in the environment for executing the activities, is available for the following design phases.

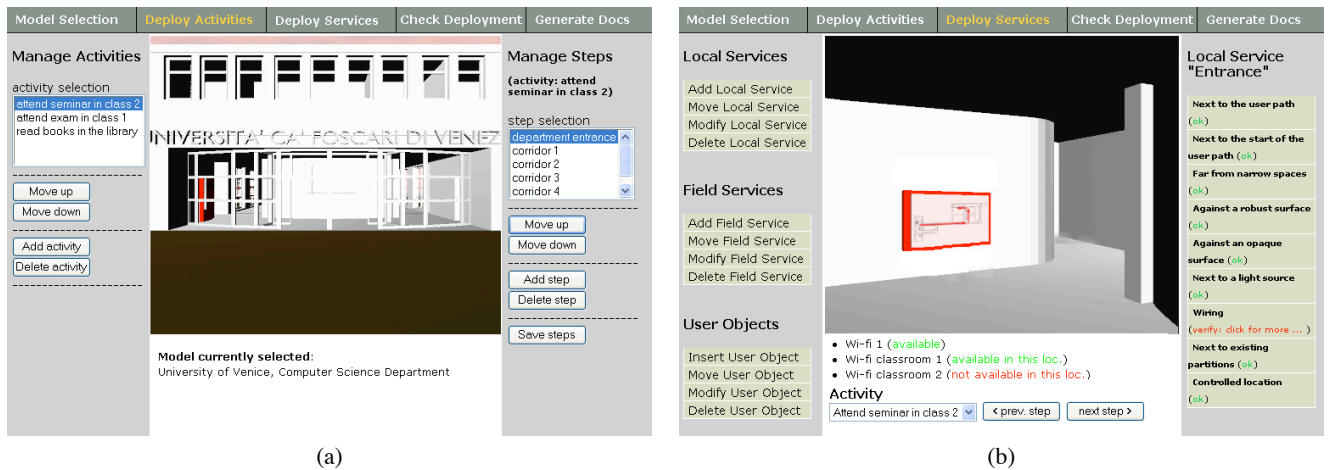


Figure 4: The visual interface: (a) deploying activities, (b) deploying services

Figure 4.b displays a snapshot of the service deployment phase. The designer selects one of the activities defined in the previous phase, using the menu displayed on the bottom of the screen. This choice determines the initial view on the 3D window, according to the designed activity deployment. The designer moves along the path associated to the chosen activity using the buttons placed next to the menu, and can also decide to depart from the predefined path and to move freely in the environment. The designer, in such way, can simulate both the behavior of an expert user inside the real environment and of a user who doesn't know which services are available and is only able to identify them when they are visible.

While local services are made perceivable by sight, the field services associated to the environment are listed on the bottom of the interface because they can correspond to visible or hidden service points. Their availability in relation to the current user location is notified as a label next to the service item.

The column on the left enables the designer to add and remove local services and to move them to reach a specific location. The designer may also choose to add field services and define their properties, such as the coverage area.

The column on the right displays the list of constraints associated to the currently selected local service, in order to allow the designer to check the consistency of the deployment step-by-step, individually analyzing each service.

The example displayed shows the *Entrance* location where a local service with the same name is placed: an interactive panel delivering information about classroom location and current use. The list of the constraints associated to the category of devices classified as *kiosk* is displayed on the right. For each constraint the constraint verifier checks the consistency of the solution. If any problem is found (e.g., the placement of the local service next to the entrance is not compatible with the building wiring), a warning notifies the designer to manually check the item in order to find an alternative design.

If the result of the checking activity doesn't evidence any problem, the designer can update all the documents related to the new solution (i.e., the list of the service instances on the 2D building map) by clicking the button *Generate Docs*.

In many situations a given environment can be temporarily populated with additional services that are targeted to specific situations, such as an exhibition or a scientific conference, which are not part of the environment purpose, and need not to be accessed permanently. In such cases, without performing the complete design pro-

cess, the designer can place and move new service appliances (if they are known to the system) inside the scene. During the validation phase that will follow the modifications operated by the user the system will consider also the presence of such new appliances and their additional constraints.

## 6. CONCLUSION

Deploying application services in complex environment requires architects, interior designers, application engineers and interaction designers to coordinate their effort to select proper service placements according to the environment functions and to the users requirements. Good appliance deployment is one of the main goals in the architectural domain, often achieved mainly thanks to the designer experience. Design systems based on formal grounds could provide a set of guidelines easier to apply and verifiable.

The use of knowledge based descriptions of activities, services and ambients constitutes a further step towards the design of interactive systems in pervasive environments. Existing environments could be insufficient to fulfill the trend of pervasive applications. The visual approach proposed in this paper can assist the designer to simulate environments and to move inside them with a VR-based metaphor to find nearly optimal placement of interactive appliances.

## 7. REFERENCES

- [1] Extensible 3D (X3D) ISO/IEC 19775:2004. <http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification>, 2004.
- [2] O. Bertelsen and S. Bodker. Activity theory. In J. Carroll, editor, *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, pages 291–324, San Francisco, 2003. Morgan Kaufmann.
- [3] A. Celentano, A. Okroglic, and F. Pittarello. An ontology based approach to interaction ambient design. In *Proc. Workshop on Mobile Services-oriented Architectures and Ontologies (MoSO)*. IEEE Computer Society, 2007.
- [4] F. Pittarello and A. D. Faveri. A semantic description of 3d environments: a proposal based on web standards. In *Proc. Web3D 2006*, pages 85–95, 2006.
- [5] M. Vukovic and P. Robinson. Adaptive, planning-based, web service composition for context awareness. In *Proc. Int. Conf. on Pervasive Computing*, 2004.