Designing Pervasive Services for Physical Hypermedia

Cecilia Challiol, Silvia Gordillo, Gustavo Rossi and Robert Laurini

Abstract—In this paper we describe the design and implementation of a software substrate for building pervasive services in the context of physical hypermedia applications. We first introduce the main ideas behind physical hypermedia; next we argue that physical navigation requires some software support to improve accessibility to real world objects. We next describe an architectural framework that supports specification and deployment of pervasive services. Some simple examples of use are presented. We conclude by comparing our work with others' and describing further work we are pursuing.

Index Terms—Mobile Computing, Context-Aware Computing, User Interfaces and Interaction Models

I. INTRODUCTION

PHYSICAL hypermedia applications [5] unify several well known types of computer software such as ambient computing, location-aware and pervasive software, by using the well-known hypermedia paradigm. A physical hypermedia application is a hypermedia application (i.e. access to information is provided by navigation as in the Web), in which all or some of the objects of interest are real world objects and links might be conventional (e.g. as Web links) or "physical", therefore requiring the user to change his position to access the target object.

A simple example might be a mobile tourist guide. When the user is in front of a tourist attraction (a church, monument, square, etc) which is recognized by the application, he gets additional information on the attraction in his mobile device. This information, which can be interpreted as the response to an implicit request triggered by the user's position, is considered as the digital counterpart of a physical hypermedia node (the tourist attraction); it may contain links to other digital documents, as a "conventional" web page, and it may exhibit references to other physical objects (a related monument, another church, etc) also in the form of hypermedia links. When the user selects one of these latter (called physical) links, he receives a map or plan showing how to reach the corresponding real-world object. If he decides to do so, he "walks" the link [7], and when he faces the new object the cycle begins again. However, during his walk, he might instead choose to visit other places or just browse through digital information: navigation might be interrupted.

We have developed a modeling approach for physical hypermedia applications [4]. Our model based approach consists of three implementation-independent stages: Conceptual, Navigation and Interface modeling. During conceptual modeling, application objects are described without taking into account navigation features; physical objects are specified as roles of conceptual objects [9]. The navigation model shows the hypermedia nodes the user will access and the links he will traverse. Walking links (Wlinks) express those relationships whose targets are physical objects. The interface model specifies the look and feel of the application. Finally, different implementations can be derived from these models. For example, we have built an extension to Jakarta Struts that allows deploying this kind of software in a conventional Web architecture [1]. In this paper we report our ongoing research on the development of a software framework for pervasive, physical hypermedia services. We first characterize physical hypermedia as a specific kind of pervasive software; we next discuss the problem of supporting physical navigation and propose a set of base location-aware services. We then present our support architecture; finally we conclude and give on outline of some further work we are pursuing.

II. PHYSICAL HYPERMEDIA VS PERVASIVE SERVICES

As previously defined, a physical hypermedia application is a specific kind of pervasive software, which basically aims at enhancing real world objects with digital information and links; we can view this software as providing two abstract coarse grained services *HInformation* (which includes object's properties and links) and *Browse* which might be refined into *DigitalBrowse* and *PhysicalBrowse*, the former providing support for conventional Web-like links and the latter aims at supporting the user to reach the target object. A simplified schema of physical hypermedia and this former characterization of services can be seen in Figure 1, where we can see that physical objects have a digital counterpart; links may be digital or physical. We also indicate the two coarse grained services which the user perceives.
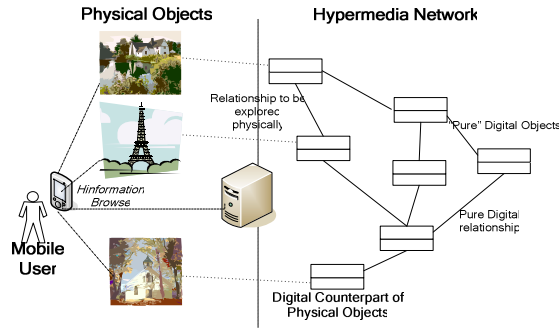
Figure 1: Enriching Real World Objects with hypermedia

As in other applications of pervasive computing, these services can be adapted to the user's needs or preferences by filtering information and links according to his profile or activity. However, the distinctive feature of physical hypermedia is the "pervasive" notion of navigation which underlies the whole user experience. However, while digital navigation has been widely explored in the literature on hypermedia and the Web (particularly for ubiquitous access), little or nothing has been said about which services are necessary to support physical navigation.

Physical or walking navigation consists in traversing the physical space with a task in mind: reaching the link target object [1]. The main difference between digital and physical navigation is that while the former is atomic (once triggered it finishes), the latter might take time and depends on the user's will, and sometimes on environmental conditions: the user may change his mind, get lost in his way, decide a detour, etc. Our aim is to provide a set of services to improve this task.

We have developed a conceptual framework for the definition and provision of pervasive services to support physical navigation. The framework has been inspired in [10] which defines a taxonomy of object roles for providing support to the handicapped user in the WWW. This taxonomy is based on a characterization of real world objects from the point of view of a traveler; we combined the original idea (real world) with the adaptation in [10] (digital world), in order to provide digital help for physical navigation. The framework provides a set of predefined roles which can be assigned to physical objects and a model of user navigation through the physical space. A physical object can for example play the role of Navigation Point (such as a street), Identification Point (e.g. the name of the street), Alert (a traffic light or a sign), Reference Point (a landmark), etc [10]. When playing a role (e.g. Alert) a physical object exhibits a set of meaningful services, which might be different from those provided by the same object when viewed in another situation (i.e. playing another role). All physical objects that the user faces during his detour will also provide a set of common services such as: *CancelNavigation*, *MapUpdate,* etc, which are fundamental for the trip. Besides, some of the intended services can be provided by different roles.

---

[1] Notice that the user might also have a "higher-level" task such as acquiring information about some topic, repairing an equipment, buying some goods, etc.

Using a role taxonomy has two benefits: it lets the designer decide which services must be provided in a specific application and situation, either by refining the taxonomy or adding new services, and it also helps to avoid overwhelming the user by prioritizing services according to the intended physical object's role.

Roles are assigned to physical objects dynamically according to the current user's state regarding navigation. While the user is traversing the physical space to reach an object, he may interact with intermediate objects to receive assistance in his task. For example suppose that while facing an object X, he chooses a link to an object Y. Using the map, he walks to the target (figure 2); knowing that this is the current user's task the intermediate objects will play the intended role. For example the object Z in Figure 2 will play the role of navigation point. As a consequence at the same moment an object (e.g. Z) might be playing two different roles (e.g. for different users), or even more interesting twice the same role but indicating different paths (See also Figure 6). In Figure 2 we show this situation. Schematically, while the user A is pursuing a travel to object Y, user B is walking towards Z. In this way, when they face Z, the object behaves differently (playing different roles) for each of the users.
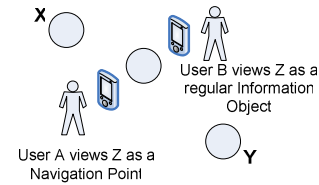

Figure 2: Different users face different roles of the same object

III. AN ARCHITECTURE FOR PERVASIVE HYPERMEDIA SERVICES

We have implemented our conceptual framework on top of a software architecture for context-aware services. Further details about the general architecture can be read in [2].

A. Details of the Architecture

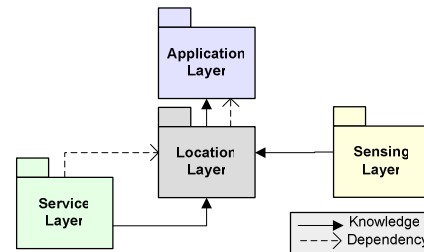The general schema of the architecture is presented in Figure 3.

Figure 3: High-Level View of our architecture

In the Application layer, we specify application (digital) objects and behaviors in such a way that they are oblivious with respect to their physical nature or their eventual location features. Therefore, the same application model can be used for building "conventional" Web applications (i.e. in which

objects' locations are not important). In our example we will specify classes which represent the Tourist attractions and their relationships. If we aim to provide selling services (as in Museums' boutiques for example) we will also have classes for the boutique, the behaviors for selling, etc.

Some of these objects will have a physical presence and we wish that the user can be sensed when he is in front of them; therefore we need to record the objects' positions. The information regarding objects and actual user's location is specified in the Location layer. This layer also contains those physical objects, which even not being relevant for the underlying application, are meaningful in the physical world such as streets, corridors, stairs, semaphores, advertisement walls, etc. By decoupling the physicality of an object from its other properties we improve systems' evolution; we also allow to provide physical presence at the object (instead of class) level (e.g. some monuments are accessible, others are not or do not exist anymore though we still have their digital information). We also separate physical objects from their location representation to get independence from the reference system (latitude/longitude, symbolic, geometric, semantic, etc).

The Sensing layer (not described in detail for the sake of conciseness), allows to bridge hardware abstractions (e.g. sensors) from the higher level representation of physical objects. Finally, the Service layer contains the (location-aware) services that the system provides. Services, which are first class objects, are attached to locations by using the concept of Service Area. This layer also has the needed machinery to determine which services must be activated in a given moment by also representing the services to which a particular user is subscribed (allowing that a user can choose which services he likes to use).

The architecture uses dependency (publish/subscribe) mechanisms to trigger user services. When the user changes his position (which is determined at the Sensing layer), the object representing his position is updated triggering a message to the Service Environment, which in turn determines in which Service Area the user is located and thus which are the current available services.

We have used this architecture to extend legacy applications (described in the Application layer) with pervasive, location-aware services. It provides a simple way to seamlessly add physical presence to application objects; it supports different ways of sensing the user's position and it gives a modular mechanism for assigning services to physical areas. We next describe how we enhanced the architecture for providing physical hypermedia services.

B. Architecting Physical Hypermedia Services

As explained in Section II, physical hypermedia services do not execute in a vacuum or just taking into account the user's position and preferences. His current navigation task and history is critical to decide which services will be provided; we call these services, navigation aware services. The first enhancement to the base architecture is to enhance physical objects with a default service, *HInformation* (standing for hypermedia information) which gives them the semantics of

hypermedia nodes. *HInformation* returns the equivalent to a Web page by querying the intended object(s). Notice that as explained below, the actual service(s) that the user has available might change according to the current user's state and therefore according to the role assigned to the physical object he is facing.

A digital link is traversed as in a Web application (in fact the information is displayed in a browser); meanwhile, when the user selects a WLink, he is provided with information to reach the target object; this information can be customized by the designer, either supplying corresponding maps or plans, or adding some specific components to find routes in a city.

While the user traverses the physical space different services are provided. As previously mentioned our framework comprises an open set of role classes, which exhibit predefined services. Both the role hierarchy and services can be extended or modified for a specific application need. Roles can be attached to physical objects to let them provide additional services. A class diagram showing this service micro-architecture is shown in Figure 4.
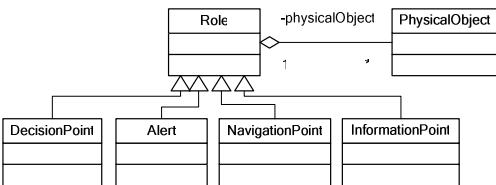

Figure 4: Enhancing Physical Objects with Roles

At configuration time the designer decides which roles can be attached to each physical object; e.g. we might not want that a Church plays the role of Alert or we might want that in certain cases an Advertisement plays the role of Navigation Point. We also assign navigation states to corresponding roles, e.g. while the user is in state X, certain objects must exhibit role Y.

The user navigation state is represented as an instance of the State pattern [3] as partially shown in Figure 5; the navigation history is recorded as a list of traversed objects and navigation states. In a specific application, the designer may extend the State hierarchy if needed.
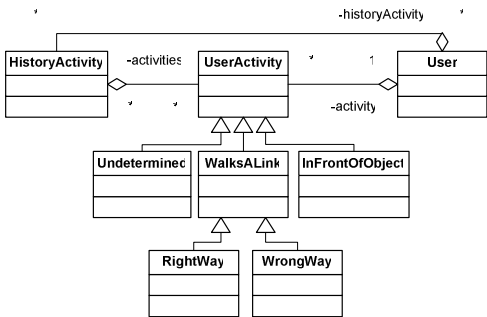

Figure 5: User navigation state and history

When the user is sensed to be facing a physical object, the object representing the current user's state (e.g. an instance of WalksALink) acts as a Role builder, creates the corresponding role (provided that the object can be configured with the role),

attach it to the physical object and the intended services are presented to the user.

As a proof of concept we have developed a prototypical application in a Natural Science Museum. The physical objects are skeletons of pre-historical animals, which have been enriched with simple digital information and hypermedia links. The prototype uses a HP iPaq 2210 with infrared as sensing hardware. A simple example of how the schema in Figure 2 works according to the previous explanations can be seen in Figure 6. The user A is moving from Velociraptor to Tiranosaurus; when he faces Herrarosaurus, the object plays the role of Navigation Point, indicates that the user is the correct way and offers some additional services, one of which is to view the Herrarosaurus's information. If the user selects this service the information will not show additional physical links except that the user cancels his actual navigation to Tiranosaurus. Meanwhile, User B is not traversing any physical link and therefore when he faces Herrarosaurus, the default *HInformation* service is triggered showing the hypermedia information shown in Figure 6 at the right.
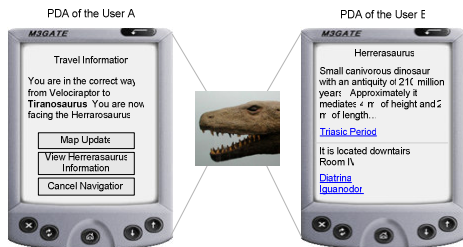

Figure 6: Two different users facing the same object while performing different navigation tasks

IV. RELATED WORK

In [6] a comprehensive framework (HyCon) for deploying applications in which the hypermedia paradigm is extended to the physical world is presented. This framework provides help to the navigator by recording and managing the user's context. Our research has been somewhat inspired in this seminal work, though we chose to extend a service-oriented architecture with hypermedia services. In [8] meanwhile, an object-oriented framework called HyperReal, based on the Dexter hypertext reference model is presented. Differently from these two approaches, our work is grounded in a model-based approach: we explicitly represent the objects of interest and their relationships in an application (or conceptual) model and base the hypermedia links (digital and physical) in a navigational model. We also use the concept of travel objects roles to better characterize and organize pervasive services. Finally, in [2] we have presented a software architecture for providing context-aware services. Our contribution in this paper is the definition of a new kind of context (the travel context), a support for hypermedia navigation and an open set of physical hypermedia services.

V. CONCLUDING REMARK AND FURTHER WORK

In this paper we have presented our approach for building pervasive services for physical hypermedia applications. We showed why these kinds of services are useful, and gave an example of the kind of services that may be provided according to the role that a physical object plays. We also show how to determine this role taking into account the current user's activity. We have briefly described how we extended a service-oriented architecture for location-aware applications with the notion of navigation activity; our framework provides an open set of travel object roles which can be dynamically assigned to physical objects according to the actual state of the user in his navigation task. As the framework is open, different kinds of applications can be built on top of it; designers can refine the existing abstractions either with domain-specific behaviors, location models, etc. We are currently improving the framework in order to integrate pervasive services with information provided by third parties, e.g. via Web services. In that way we can enrich the user's navigation experience either with environmental information (weather services) or with additional attractions that fit the user's preferences. We are also studying usability issues to improve services interfaces.

REFERENCES

1. Challiol, C., Rossi, G., Gordillo, S., De Cristofolo, V.: Designing and Implementing Physical Hypermedia Applications. In Proceedings of the 2006 Workshop on Ubiquitous Web and System Intelligence (UWSI 2006), Springer Verlag, LNCS, forthcoming.
2. Fortier, A., Rossi, G., Gordillo, S.: Decoupling Design Concerns in Location-Aware Services. In Proceedings of the IFIP Conference on Mobile Information Systems (MOBIS), 2005, Springer, 2005.
3. Gamma, E., Helm, R., Johnson, J., Vlissides, J.: Design Patterns. Elements of reusable object-oriented software, Addison Wesley 1995.
4. Gordillo, S., Rossi, G., Schwabe, D.: Separation of Structural Concerns in Physical Hypermedia Models. CAiSE 2005: 446-459
5. Gronbaek, K., Kristensen, J., Eriksen, M.: Physical Hypermedia: Organizing Collections of Mixed Physical and Digital Material. Proceedings of the 14th. ACM International Conference of Hypertext and Hypermedia (Hypertext 2003), ACM Press, 10-19.
6. Hansen, F., Bouvin, N., Christensen, B., Gronbaek, K., Pedersen, T., Gagach, J.: Integrating the Web and the World: Contextual Trails on the Move. Proceedings of the 15th. ACM International Conference of Hypertext and Hypermedia (Hypertext 2004), ACM Press. 2004.
7. Harper, S., Goble, C., Pettitt, S.: proXimity: Walking the Link. In Journal of Digital Information, Volume 5, Issue 1, Article No 236, 2004-04-07. http//jodi.ecs.soton.ac.uk/Articles/v05/i01/Harper/.
8. Romero, L., Correia, N.: HyperReal: A Hypermedia model for Mixed Reality. Proceedings of the 14th ACM International Conference of Hypertext and Hypermedia (Hypertext 2003), ACM Press, 2-9.
9. Steimann, F.: On the Representation of Roles in Object-Oriented and Conceptual modeling. Data and Knowledge Engineering 35 (2000) 83-106.
10. Yesilada, Y., Stevens, R., Goble, C.: A foundation for tool based mobility support for visually impaired web users. In Proceedings of the Twelfth International Conference on World Wide Web, pages 422–430, 2003.