Modeling Physical Hypermedia Applications

Silvia Gordillo[1,3], Gustavo Rossi[1,4], Fernando Lyardet [2]

[1] Lifia. Facultad de Informatica. Universidad Nacional de La Plata, Argentina
[2] Fernando D. LyardetTelecooperation, Technische Universität Darmstadt, Germany
[3]Also at CICPBA, [4]Also at Conicet
{gordillo,gustavo}@lifia.info.unlp.edu.ar, fernando@tk.informatik.tu-darmstadt.de

Abstract

In this paper we present a modeling approach for building physical hypermedia (PH) applications, i.e. those applications in which real and digital objects are navigated using the hypertext paradigm. We first survey the state of the art in this kind of ubiquitous applications. We stress the importance of modeling and design activities and present our design approach, an extension of the well-known Object-Oriented Hypermedia Design Method (OOHDM); we finally discuss some further work we are pursuing in this area

1. Introduction and Background

In this paper we are interested in a novel way of applying wireless and location-sensing technologies to support knowledge and information management in cyberspace: the paradigm of physical hypermedia (PH). In these software systems we can build meaningful relationships among physical, real-world objects and virtual (digital) ones by using the hypermedia information model.

The relationships between physical and digital objects and the different ways for augmenting the real world using computing devices have been early discussed in [6]. In [1] the authors have shown how to add digital annotations (such as graffiti) to physical places and objects, and the social impact of their use and sharing. In [7] the authors introduced a hypermedia model for dealing with mixed reality

The idea of PH was first introduced in [3] and refined in [4] as a formalism to build augmented reality applications. In PH, physical objects are augmented with digital information which can be accessed by the mobile user, for example while being in front of the object. Objects are further considered as nodes in a hypermedia network and thus linked with other nodes (either physical or digital). In some cases the hypermedia link is traversed using the well-known navigation paradigm (e.g. as in the WWW), while in other situations the link has to be "walked" by the user [5].

A simple example application scenario may be a Museum in which visitors are equipped with portable computer devices, and there is some location sensing mechanism. When the visitor stands in front of an artwork, he can see its digital representation. Additionally he is presented with a set of anchors that allow him to navigate to other nodes (information items) related with the artwork. When one of these nodes represents a physical object, he is informed on how to reach that object (perhaps another artwork); then, he can choose to traverse the physical space ("walk" the link) towards this node or just continue his tour. Notice that we are not just augmenting the physical object (artwork) with some digital information but also providing some kind of linking to other digital or physical objects. This paradigm is appealing because as shown in [1], it can provide the basis for interesting social interactions, such as people leaving comments, personal impressions, etc. As with "conventional" hypermedia and its usual implementation in the web, we can further use the paradigm to model and implement applications in areas such as mobile-commerce, e-government, etc.

In this paper we present a novel approach to model PH applications; this approach extends the Object-Oriented Hypermedia Design Method (OOHDM) [9] by adding some new abstractions and re-defining the semantics of basic navigational behaviors. Our contribution is twofold: first we indicate which design issues must be faced while modeling PH applications, and second we present a simple solution that can be easily adapted to other similar design approaches.

The rest of the paper is organized as follows: In Section 2, we introduce our approach; we briefly explain the OOHDM framework and indicate how we extended it to support PH. Finally, we present some further work we are pursuing.

0-7695-2263-7/05 $20.00 © 2005 IEEE

2. The Design Approach

While researchers have emphasized the feasibility of the PH paradigm by building software infrastructures that support these ideas [3,4,5], modeling and design issues have been so far neglected. We are interested in these aspects because they are a mean to deal with this kind of software, in an implementation-independent way; besides, while modeling we can easily incorporate users and other (non-software) specialists in the development process, thus improving the quality of the resulting system.

To make this discussion concrete, we define a PH application as a hypermedia application (i.e. the access to information objects is done by navigation), in which all or some of the objects of interest are real-world objects which are visited by the user "physically". We assume that in a PH application there is some underlying location-sensing technology that allows the application to be aware of the actual user's position.

In these applications we can see two different ways to implement hypermedia navigation: the usual way (i.e. selecting an anchor and accessing the target object), and the physical way in which the user must change his position and walk towards the object.

As with other kind of software, we have to specify unambiguously its intended structure and behavioral semantics. We must express, in an implementation-independent way, which are the objects of interest and their properties (including their location), how they are linked, which links should be implemented as conventional, and which should be "walked" by the user. A solid design approach should be able to cope with technology evolution and heterogeneity, i.e. the design model should not be compromised with details on location-sensing technology, and at the same time it should allow to build models that can gracefully evolve together with new requirements and/or technological possibilities.

We have extended the OOHDM [9] design approach by adding some new concepts such as physical objects and slightly changing some hypermedia navigation semantics to adapt them to the physical hypermedia field.

2.1. The Basic Design Framework

OOHDM partitions the development space into four activities: conceptual modeling, navigation design, abstract interface design and implementation. During conceptual modeling we describe the application classes and their relationships using UML [10]. The focus is put on generic application's behavior and the application is modeled neutrally with respect to navigation issues. In OOHDM, a hypermedia or Web application is seen as a navigational view over the conceptual model, and we can specify different views according to the user profile or role.

For each user profile we can define a different navigational structure, which will reflect objects and relationships in the conceptual schema according to the tasks this kind of user must perform. The navigational structure of a Web application is defined by a schema, containing navigational classes such as nodes, links, anchors and access structures.

The semantics of nodes, links and anchors are the usual in hypermedia applications. Access structures, such as indexes, represent possible ways for starting navigation.

The abstract interface model defines which interface objects the user will perceive (in particular how nodes will look like) and which interface transformations will take place. Finally, during implementation the whole set of models is mapped into a run-time environment. Though OOHDM does not prescribe a particular strategy for implementing a hypermedia or Web application, the design style facilitates the use of object-oriented languages and architectural styles such as the Model-View-Controller metaphor. In Figure 1 we summarize this process indicating design "hot-spots" where "physical" aspects should be added. In the following sub-sections we will focus on conceptual and navigational aspects.

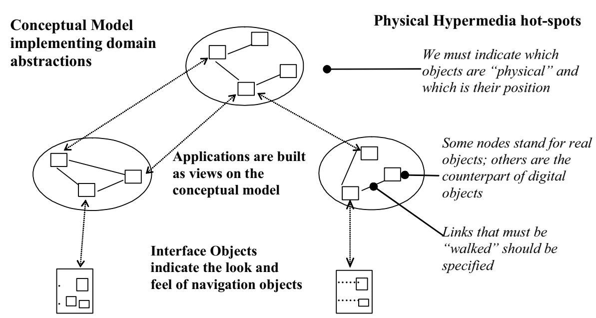

Figure 1: The OOHDM design space

2.2 Dealing with physical Objects

We extended the OOHDM meta-model by adding the concept of Physical Object. A physical object is an application object that can be explored "physically", i.e. it will have a physical presence in the system, and we can sense if the user is near it. In the museum example, we can be interested in modeling artworks and even rooms as physical objects. To find a suitable

0-7695-2263-7/05 $20.00 © 2005 IEEE

approach for modeling physical objects, we need to consider that not all objects in a class (e.g. Artwork) must be tagged as physical: for example, we might want to relate artworks that exist physically with others that are not in exhibition, are in another geographical place or simply do not exist anymore.

Representing physical objects as sub-classes of a particular class (Artwork) introduce a specialization criteria that might collide with others in the intended domain (paintings, sculptures, etc). We have chosen to model physical objects as roles that can be assumed by conceptual objects [8]. A role type (in this case "physical") indicates those properties and behaviors of an object when playing that role. Roles can easily be mapped to implementation settings using for example decorators as shown in [2].

In Figure 2 we show a simple conceptual model for the physical museum; roles are described using the notation in [8] that extends UML with roles. Artworks, Rooms, Boutiques and the Museum itself are physical objects; thus there is a specific role type for each of them. We only show some of the conceptual relationships for the sake of conciseness. The Physical role hierarchy (not shown here) describes features of every role type that can be played by a physical object.

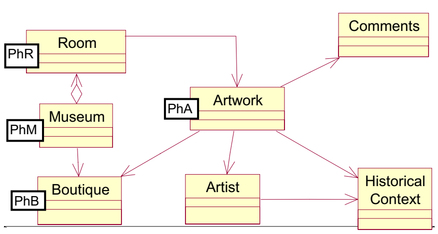

Figure 2: Conceptual Model including physical objects

Physical objects are characterized by an attribute *position* whose semantics depends on the location sensing technology, so it must be refined for each application. Different role types (e.g. Museum and Boutique) might use different ways of location sensing and representation: for example, if we use infrared technology, *position* might be implemented using just an identifier, meanwhile, in outdoor applications that use GPS or other sensing techniques, *position* must be implemented using more complex location models. Physical objects implement the *inFrontOfMe (user)* behavior that is triggered by the underlying software when the user is sensed to be in the object's vicinity. The standard behavior is to open the corresponding node (See 2.3). Additionally they should implement the *howToReachFrom (location)* which is used by walking links (See 2.3) to indicate how the user can find the object.

Clearly, separating the conceptual from the physical aspect of an object allows to decouple design decision and to build different browsing strategies according to the dimension we are considering (e.g. physical or digital)

2.3 Specifying Navigation aspects

The navigation schema shows which nodes the user will perceive and which links he can follow. Nodes are built from conceptual objects and links are derived from relationships in the conceptual model. One of the cornerstones of OOHDM is that a different navigation schema can be built for different user roles.

In the Museum application, we can for example build a different navigation schema for the regular visitor or for an expert (for example a person working in the Museum). Some artworks might be even (physically) inaccessible for a visitor, while the museum worker should be able to access them for performing his work.

There are two important differences between a conventional and a physical hypermedia regarding the navigational schema: the activation of nodes and the semantics of link traversal. In conventional hypermedia a node is opened when we navigate a link having that node as a target.

While we want to preserve this behavior for "pure" digital nodes, a node that stands for a physical object should only be opened when the user is facing the object. We decided not change the basic Nodes class hierarchy, but instead introduced some changes in the physical objects (role) classes and in the link class behaviors.

Regarding objects, as we explained in Section 2.2, a node that express a "digital" view of a physical object will be opened when the user is in front of the object.

To implement a different navigation semantics we defined, walking links (or WLinks) as those links whose target node is the digital counterpart of a physical object. The main difference between the operational semantics of a navigational and a walking link is that while the former closes the current node and opens the target node, the latter just indicates the user intention to reach the corresponding physical object. In order to achieve this, the link invokes the *howToReachFrom* behavior in the physical object corresponding to the target node, using as a parameter the current user location. The answer might vary for different applications (and implementations): it may show a proposed itinerary, a plan with the position of the physical object, etc. In our design framework we leave this decision to the designer, who can specify and implement these or other strategies.

WLinks are designed by changing the default link traversal algorithm, which in OOHDM is expressed as a Strategy [2] on Link classes as described in [9]. Thus, even different types of "walking" semantics can be expressed by specifying another algorithm (this discussion is outside the scope of the paper).

Decoupling links from their traversal algorithms also allows us to express differences at the link instance level, for example when an instance of a WLink class has a "non-walking" semantic, i.e. it behaves as a "conventional" link. In Figure 3 we show the navigational schema for the visitor user role that corresponds to the conceptual model in Figure 2. We have eliminated the Historical Context for visitors, which is possible in OOHDM when defining views.

In Figure 3, we show WLinks with a <<W>> in the style of UML stereotypes [10]; as said before we can have "non-walking" instances of a WLink simply by specifying it at the instance level.

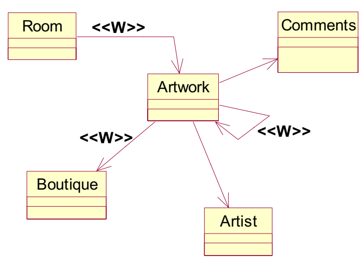

Figure 3: Navigational Schema with WLinks

3. Discussion and Further Work

In this paper we have introduced a modeling approach for physical hypermedia applications. We have shown that using well-known software engineering practices we can specify the more relevant aspects of an application that integrates the real and the digital worlds using the hypermedia paradigm.

Our approach differs from others in the literature [3,5,7] in that it emphasizes modeling with respect to infrastructure or implementation.

We are currently working in several research lines; first we are improving our notation to make it more "standard" by using UML extension mechanisms and defining stereotypes for physical objects and WLinks.

We are also studying the integration of different modeling dimensions in the same conceptual model. For example, there may be relationships between physical objects that go beyond the conceptual ones, such as those involving proximity, vicinity, or other spatial relationships. It is interesting to analyze how these relationships might influence the navigation schema, i.e. in which way the user can choose to follow spatial relationships, even though they do not contain strong application semantics. We are finally evaluating how to incorporate a complete user model to specify customized and context-aware navigation styles.

4. References

[1] F. Espinoza, P. Persson, A. Sandin, H. Nystrom, E. Cacciatore, M. Bylund: "GeoNotes: Social and Navigational Aspects of Location-Based Information Systems". Proceedings of Third International Conference on Ubiquitous Computing (Ubicomp 2001), pp 2-17. Springer Verlag.

[2] E. Gamma, R. Helm, R. Johnson, J. Vlissides: "Design Patterns. Elements of Reusable Software", Addison Wesley 1994.

[3] K. Gronbaek, J. Kristensen, M. Eriksen: "Physical Hypermedia: Organizing Collections of Mixed Physical and Digital Material". Proceedings of the 14th. ACM International Conference of Hypertext and Hypermedia (Hypertext 2003), pp 10-19, ACM Press.

[4] F. Hansen, N. Bouvin, B. Christensen, K. Gronbaek, T. Pedersen, J. Gagach: "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.

[5] S. Harper, C. Goble and S. Pettitt: "proximity: Walking the Link". In Journal of Digital Information, Volume 5, Issue 1, Article No 236, 2004-04-07. Available at: http//jodi.ecs.soton.ac.uk/Articles/v05/i01/Harper/

[6] W. Mackay: "Augmented Reality: Linking real and virtual worlds. A new paradigm for interacting with computers. Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI 98), ACM Press, pp13-21.

[7] L. Romero, N. Correia: "HyperReal: A Hypermedia model for Mixed Reality". Proceedings of the 14th ACM International Conference of Hypertext and Hypermedia (Hypertext 2003), pp 2-9, ACM Press.

[8] D. Riehle, T. Gross. "Role Model Based Framework Design and Integration." Proceedings of the 1998 Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA '98). ACM Press, 1998. Page 117-133

[9] D. Schwabe, G. Rossi: "An object-oriented approach to web-based application design". Theory and Practice of Object Systems (TAPOS), Special Issue on the Internet, v. 4#4, pp.207-225, October, 1998.

[10] The Unified Modeling Language: www.uml.org