DOCUMENT RESUME

ED 482 159 IR 058 823

| | |
|---|---|
| AUTHOR | Garzotto, Franca; Salmon, Tullio; Pigozzi, Massimiliano |
| TITLE | Designing Multi-Channel Web Frameworks for Cultural Tourism Applications: The MUSE Case Study. |
| PUB DATE | 2003-00-00 |
| NOTE | 24p.; In: Museums and the Web 2003: Selected Papers from an International Conference (7th, Charlotte, NC, March 19-22, 2003); see IR 058 801. |
| AVAILABLE FROM | For full text: http://www.archimuse.com/mw2003/papers/garzotto/garzotto.html/. |
| PUB TYPE | Reports - Descriptive (141) -- Speeches/Meeting Papers (150) |
| EDRS PRICE | EDRS Price MF01/PC01 Plus Postage. |
| DESCRIPTORS | Access to Information; *Computer Interfaces; Computer System Design; Culture; Foreign Countries; Information Technology; Museums; *Tourism; World Wide Web |
| IDENTIFIERS | Italy |

ABSTRACT

A framework for the design of multi-channel (MC) applications in the cultural tourism domain is presented. Several heterogeneous interface devices are supported including location-sensitive mobile units, on-site stationary devices, and personalized CDs that extend the on-site experience beyond the visit time thanks to personal memories gathered during the visit. The design framework is multi-layer in nature: it takes care of application design both at conceptual and implementation level. At conceptual level it supports content, navigation, interaction and presentation design. At implementation level it includes an interface independent execution engine as well as a set of tools mapping the design into a formal interface description that specifies the run-time rules-of-behavior to the execution engine. The proposed framework is going to be demonstrated in museums and archaeological sites in Italy. Includes 10 figures. (Contains 21 references.) (Author)

PAPERS
Museums and the Web 2003

# Designing Multi-Channel Web Frameworks For Cultural Tourism Applications: The MUSE Case Study

Franca Garzotto, Politecnico di Milano; Tullio Salmon Cinotti, Università di Bologna; and Massimiliano Pigozzi Casalecchio di Reno, Bologna, Italy





A&MI

Archives & Museum Informatics
158 Lee Avenue
Toronto Ontario
M4E 2P3 Canada

ph: +1 416-691-2516
fx: +1 416-352-6025

info@archimuse.com
www.archimuse.com



Updated: March 13, 2003

## Abstract

A framework for the design of multi-channel (MC) applications in the cultural tourism domain is presented. Several heterogeneous interface devices are supported including location-sensitive mobile units, on-site stationary devices, and personalized CDs that extend the on-site experience beyond the visit time thanks to personal memories gathered during the visit. The design framework is multi-layer in nature: it takes care of application design both at conceptual and implementation level. At conceptual level it supports content, navigation, interaction and presentation design. At implementation level it includes an interface-independent execution engine as well as a set of tools mapping the design into a formal interface description that specifies the run-time rules-of-behavior to the execution engine.

The proposed framework is going to be demonstrated in outstanding museums and archaeological sites in Italy.

***Keywords:*** *multi-channel applications design, pervasive computing, cultural tourism, context-awareness, zero-interface.*

### *Introduction*

The Information Society expects the establishment of pervasive services (Abowd 1999, Weiser 1993) providing friendly and effective *education* to everybody, in every place, at any time, improving the *mutual understanding* between people of different countries, and impacting our approach to *global culture* Cultural tourism offers a precious opportunity to verify the potential and the impact of this emerging technological scenario. The approach of today's tourists to culture has significantly changed with respect to the past. The primary goal of the travellers in the XVII century was "seeing" and "wondering". For travelers of the third millennium, "learning" is usually one of the main objectives.

Information technology offers new challenges for the creation of an education environment that is based on cultural heritage and addresses the *need for knowledge* of cultural tourists (Oppermann 1999, Bocchi 1999, Guidazzoli 2000). In particular, the technology of *multichannel Web applications* (MC applications for short) promises a successful approach to these goals. An MC application delivers its services on several stationary or mobile devices or "channels"; such as desktop PCs, notebooks, mobile phones, PDAs, Web TVs and dedicated appliances. An MC application relies upon a centralized application environment which manages a shared information base coupled to the so-called "business logic" ( Salmon Cinotti 2001). On each channel, it delivers services which are *customized* to the characteristics of the delivery device and of the user context (Ozen 2001, Perkowitz 1999). As such, MC applications offer the possibility of making cultural services available on-demand wherever fruitful and convenient for a tourist, in a way appropriate to the physical and logical situation of use.

The state-of-the-art technology provides a number of hardware and software solutions for implementing MC applications. But few methodological tools exist to help developers master the complexity of *designing* MC applications.

The MUSE project - funded by the Italian government under the leadership of Ducati Sistemi S.p.A., in cooperation with Cineca, Politecnico di Milano, Sinet and the University of Bologna - addresses the above issue in a specific domain: *cultural tourism*. One of the MUSE goals is to develop a "framework" for MC cultural tourism applications. According to software engineering, a framework "...provides a reusable solution for a class of software applications that share a common set of requirements in a given domain" (Johnson 1997). A framework can be regarded as an application "skeleton", which captures the essential features of a class of applications and can be "easily" instantiated to produce (some aspects of) a specific application in the class (Fayad et al. 1997, Schwabe et al. 1999, Garzotto et al. 1999).

The MUSE framework has different levels of abstraction. At the highest level, the MUSE framework supports the *conceptual design* of MC applications for cultural tourism. The MUSE *conceptual framework* defines the user experience for the four MUSE channels. It provides the general classes of information and navigation structures that may be shared among different applications in the domain of cultural tourism, and it defines the general navigation and interaction modes by which these contents can be used on each different channel.

At the lowest level, the MUSE framework focuses on the *technological infrastructure* , by defining the general characteristics and integration features of the MUSE delivery channels.

At anintermediate level, the MUSE framework considers the *software architecture* . It defines the middleware components that support the communication between the different MUSE channels and the shared information base.

The MUSE framework, at all of the above mentioned levels, has been tested and used for the (on going) development of three MC applications for cultural tourism: one concerning the archaeological site of Pompei, the second concerning the Certosa di San Martino (a large museum located in an XVII century monastery in Naples), and the third concerning the Florentine "Museo di Storia Della Scienza" (Museum of Science History).

The development of the MUSE MC application in Pompeii is carried out in co-operation with a team of archaeologists currently working in the project "Pompei - Insula del Centenario (IX,8)". This project started in 1999 as a co-operation between the "Dipartimento di Archeologia dell'Università di Bologna" and the "Soprintendenza Archeologica di Pompei". Its primary goal is the study and restoration of a large Roman "domus" (house), called "*Domus del Centenario* ", located in the "*Insula* " (Quarter) *IX* . The MUSE MC application focuses on the *Domus del Centenario* , on its insula, on two routes, both including the magnificent forum, which connectss the domus to the site main entrance and one of the beautiful houses located along these roads. The application contents include maps, audio-video material, images with their descriptions, 3D static and interactive models, and virtual reconstructions of large sections of *la Casa del Centenario* as it is today and as it was in 79 AD when Pompei was destroyed.

La Certosa di San Martino in Naples is a large XIV century charterhouse located on top of the Vomero Hill. It was deeply modified in the XVII century and it became "the Museo of the City of Naples" in 1867, after the abolition of all Roman Church properties ordered by the government of the United Italian Nation. The museum is partitioned into many sections, mostly located in the rooms where the Carthusian monks used to live. The monastery includes cloisters, gardens, and a church. It offers unforgettable views of the gulf, the city, and Mount Vesuvius, and is one of the most beautiful monuments in Southern Italy. The Muse MC application covers a museum section called "Immagini e Memorie della Città" ("Images and Memory of the Town") as well as the areas connecting this

section to the main entrance. The section was chosen by the museum authority because of its magnificent panorama and of its link to the city and its history. The MUSE MC application contents include text, images, and audio-video clips about paintings, historical cartography, sculptures, ceramics and furniture.

The "Museo di Storia della Scienza" in Florence is the most important History of Science Museum in Italy. It includes an outstanding collection of instruments paving the path of science evolution in many areas, including astronomy, sky physics, mechanics, optics, magnetism and electricity. The Muse MC application aims to offer an engaging education experience to communicate the meaning and the history of the museum exhibits, and in its first version will cover two halls devoted to Galileo Galilei, focusing on Galileo's instruments, his work, and his scientific achievements.

The rest of this paper will discuss the characteristics of the MUSE channels and will present the main feature of the MUSE framework.

### *Muse Channels: General Features*

The MUSE MC framework considers four different channels: the conventional Web channel, the on-site mobile channel, the on-site stationary channel, and the memories channel. The Conventional Web Channel

This channel is a conventional internet connected desktop or laptop. The user experience of an MC application on this channel is typical of any conventional cultural Web site - navigation based exploration and fruition of cultural contents- suitable for a user who is planning a trip or just wants to learn more about a cultural subject either at home or in a museum entrance or reading room.

#### The On-Site Mobile Channel

The on-site mobile channel is a context aware hand-held device intended to be used during the visit to a cultural site, e.g., a museum or an archaeological settlement. The mobile device, called Whyre, is specifically designed to act as a personal multimedia interactive guide for the visitor. Whyre, shown on fig. 1, is worn with a neck strap like a camera's. It is connected to a local server by air and it is equipped with a high brightness 6.4" display. "Whyre" enabling technology is an Intel Architecture based platform with dynamic voltage and frequency scaling developed within the Muse project with the kind support of Intel Labs.

***Figure 1: Using Whyre, the MUSE on-site mobile channel***

Due to the screen and performance limitations of the device, communication effectiveness is based on words, images, and animation. Words are mainly conveyed by audio and a limited amount of text (captions, directions, dates, or on-demand textual copy of the audio clip).

Whyre can work in two main "modes". In *Web mode* , the information base can be accessed as any standard Web site; e.g., by a visitor quietly sitting down somewhere on-site, using the conventional Web navigation paradigm. In the *context aware mode*, the device can detect the approximate user position and field-of-view in the physical space. In this mode, the application can be dynamically customized to present the multimedia content and interaction capabilities which are more appropriate to the current user position.

Context-aware customization is triggered by orientation- and location-dependent events which are identified by properly fused data originating from many on-board sensors, including a GPS, a digital compass, gyroscopes, accelerometers, a camera and a WLAN based tracker. Context-aware customization supports *serendipity* : it makes the visit full of surprises, stimulates resource discovery, stirs up questions, and provides answers at incremental depth levels. In addition, context-awareness supports the adoption of the "virtual guide metaphor": it enables the application to guide the user along site tours designed by the site curators, keeping track of where the users are in the tour, warning them in case of deviation, and suggesting how to return to the tour path.

**The On-Site Stationary Channel**

The content on Whyre is enhanced by the *on-site stationary channel* , a stationary high performance graphic station connected to a large high-resolution display located at specific points of a cultural site. The graphic station is radio-controlled by the same device used in the mobile channel, and the connection to the graphic station is automatically established when the user approaches the station's large display.

The high performance of this channel and its screen size make this channel appropriate for the delivery of cultural content where space rendering, high quality interactive multimedia, and multiple simultaneous visual presentations are key factors.

The user can access the on-site stationary channel in any place where the high-resolution display is located, and can view on the large screen channel multimedia contents that are related to that place: high quality images, 3D reconstructions and simultaneous rendering of multiple synchronised 3D models (comparing, for example, the virtual reconstruction of a building in its "original reality" and the model of its present shape). Users can interact with multimedia elements by using the Whyre control elements or by simply rolling and pitching the hand-held device: this may be appropriate, for example, to move within the 3D space and explore the virtual reconstructions. This channel supports an immersive and exciting cultural experience, and takes the visitor to an easy and engaging learning experience (Bocchi 1999, Guidazzoli 2000, Strohotte 1998). In addition, by switching from the on-site mobile channel to the on-site stationary channel, visitors can turn their individual activityinto a potential "group experience".

**The Memories Channel**

The memories channel is a multimedia CD-ROM which users can fill in during the visit and bring back home. During the on-site visit, the users can select ("mark") any cultural material presented by the on-site mobile device or by the on-site stationary device, and they can take digital pictures using Whyre to recall the emotions of the visit. As these contents are selected or created, they are sequentially stored in a Memory Album and interlinked. The result is a very

simple hypertext which provides a preview of all selected elements, a direct bi-directional link from the preview to each element and from each to the next and to the previous one. At any time, either during or at the end of the visit, the users can preview the album, add comments, delete some items, modify their order. The final album can be saved on the CD, which represents the "memory" of the on-site visit.

So far, the work of the MUSE project has focused on the three channels depicted in figure 2, since they are the most innovative and technically challenging. The conventional Web channel will be included in the framework in the future.

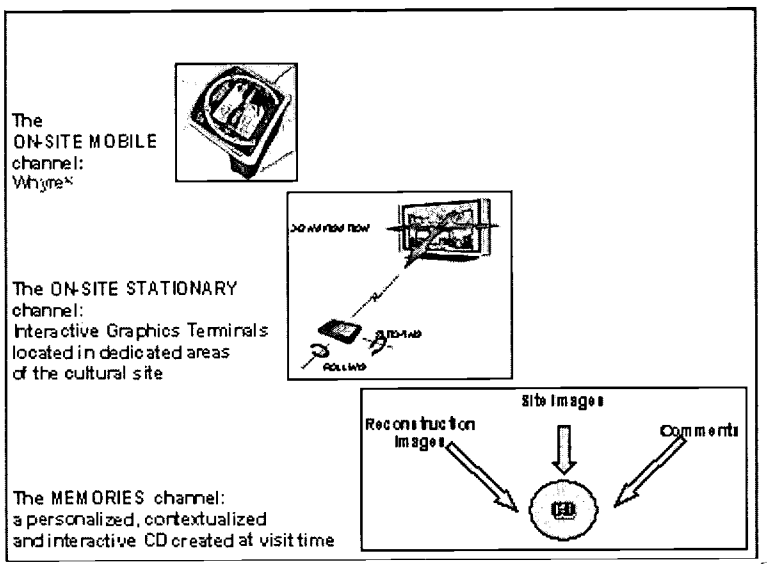


***Figure 2: Current MUSE Channels***

### *The Muse Application Design Framework*

The goal of the MUSE application design framework is to define a generalized conceptual model which captures the most relevant features of MC applications for cultural tourism and can be reused for designing, in principle, any specific MC application in this domain with a minimum amount of refinement and specialization.

The work which led to the framework definition started from the analysis of the application requirements of the three MUSE case studies: Pompei, Certosa di San Martino, and "Museo di Storia della Scienza di Firenze". Each individual case has some specificity but shares a large set of user needs. These require a common set of design solutions which have been "captured" by the MUSE application framework. To refine and empower the framework, we extended it with a number of features which we abstracted from the analysis of a large number of existing Web applications for cultural tourism.

The MUSE design framework is described using the concepts and notations of the W2000 conceptual design model. W2000, developed within the EU funded project IST - 2000 - 25131 "UWA" (Ubiquitous Web Applications), provides the modeling primitives to specify, in an implementation independent way, the various aspects of a multichannel Web application. W2000 distinguishes among the following design tasks (Garzotto et al, 1993, Garzotto et al. 1995, Baresi et al., 2000, Baresi et al. 2001):

- *Information design* : it describes the content delivered by the application. The result of information design is a set of *information design schemas.*
- *Navigation design:* it defines how contents are structured for the purpose

of navigation. The result of navigation design is a set *navigation design schemas.*

- *Interaction design* : it defines the user interaction options and the dynamic behavior of the application at run time, in response to the user actions
- *Presentation design* : it defines how navigation structures and interaction elements are presented from the delivery device. The result of presentation design is a set of *presentation design schemas* and *screen templates.*

The notation adopted in W2000 to specify the various schemas is an extension to the Unified Modeling Language UML (Booch 1998, Conallen 1999).UML is a software engineering de-facto standard that provides an extendible kit of graphical elements to model, in an object oriented fashion, data structures and behavioral properties of any software application.

A design activity which is orthogonal to the above-mentioned tasks is *customization design* . During customization design, the designer defines which application features — content, navigation, interaction, presentation — need to be specialized to the context. The term "context", in W2000, comprises all aspects concerning the situation of use: device characteristics, user profile, physical and geographical position, etc. W2000 also distinguishes between *customization at design time* and *customization at run time* . Design time customization means that the designer specifies different design schemas to address the needs of different contexts. Run time customization means that a set of adaptation rules define how some detailed design solutions are dynamically adapted to changes of context.

In MC applications for cultural tourism, both design-time customization and run-time customization are needed. Design time customization is needed to specify the content, navigation, and interaction features of the application on each different channel. Context-aware behaviour can be better expressed by using adaptation rules that define which elements must be presented on the current channel in response to user position or orientation changes.

Although MUSE framework provides general patterns for run-time adaptation rules, these are not discussed in this paper for lack of space. In the rest of this section, we will discuss issues related to design-time customization.

The problem of customizing the design to the requirements of a specific channel can be addressed in several ways:

**Approach 1: Customization starts during content design.**

Different information design schemas are produced, each one defining the content structures that are specific for each channel. A channel specific navigation schema and presentation schema are defined for each information schema.

**Approach 2: Customization is postponed to navigation design.**

The designer specifies a single information schema which defines all possible content structures for all channels. During navigation design, these are filtered and restructured for each specific channel, resulting into a set of channel-specific navigation schemas coupled with the corresponding presentation schema. In data base language, we can say that these navigation schemas define *channel specific views* on the common pool of information defined by the information schema.

**Approach 3: Customization is postponed to presentation design.**

In this case, the designers produces a single information schema, a single navigation schema, and multiple presentation schemas. The specification of

channel specific contents and links is described in the presentation structures.

Each approach has different advantages and disadvantages. Approaches 2 and 3 are more effective for authors and implementers. Since the information schema is the "summa" of all information needed for all channels, it can be regarded as an authoring check list, mentioning all content that authors must produce. For implementers, designing the "pool of contents" (i.e., the shared data base upon which the architecture of a multichannel application is built) from a single information schema is easier and less error-prone than designing it from multiple information schemas; for example, the probability of introducing information redundancies and inconsistencies is minimized. The disadvantage of approach 2 is that implementers must look at the navigation schemas in order to identify which content will be delivered to which channels. With approach 3, the implementer must inspect the page structures of each channel-specific presentation schema in order to define the content and link views that are available in each channel.

MUSE has adopted approach 2 (multiple navigation schemas); this seems to be the most effective compromise. In the following sections, we will discuss the general information schema of the MUSE framework, and we will sketch some portions of some channel specific navigation schema. Presentation schemas are omitted for lack of space, and illustrated via some screen examples. Finally, we will discuss some aspects of the interaction design.

### *MUSE Information Design Schema*

The MUSE Information Schema includes the main classes of multimedia information elements that satisfy the information requirements of an MC cultural tourism application, abstracting from the requirements of each specific channel. In principle, "any" MC application for cultural tourism can reuse this schema as a design skeleton which can then be filtered and integrated according to the needs of a specific domain.

According to W2000, two main categories of content structures are described in the information schema: *hyperbase layer structures* and *access layer structures* .

Hyperbase layer structures define the core content sources of an application. These are described in terms of *entity types* and *semantic association types* . Entity types model the main "objects" of the application domain as classes of information structures called "entities". Semantic association types model the relevant relationships among these entities.

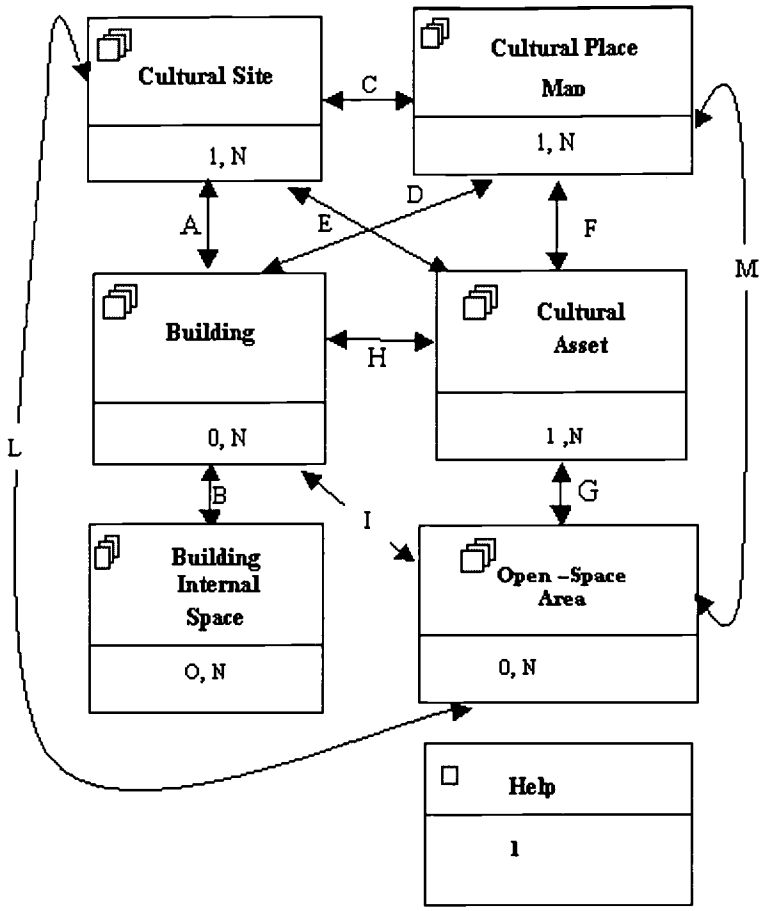


***Figure 3: The MUSE Hyperbase Information schema - in-the-large view***

Figure 3 shows that the relevant classes of information objects deal with Cultural Sites (the tourism places), their Cultural Assets, Buildings, Internal Building Spaces and Open-space areas. In addition, the application may include Maps (more or less sophisticated and rich of multimedia content) and some Help Information.

Figures 4a and 4b describe the content elements associated to the various entity types (Map Specification is omitted). Content elements within entity types are grouped in sub-structures called Component Types, characterized by a black banner. Except Cultural Place and Map, all entity types share the same content structure. In the object-oriented fashion of UML, this can be specified by defining an *Abstract Entity Type* - called *Object of Interest* in our case — and by saying that entity types Building, Cultural Asset, Building Internal Space, Open-Space Area are all sub-classes of Object of Interest (see figure 4b).

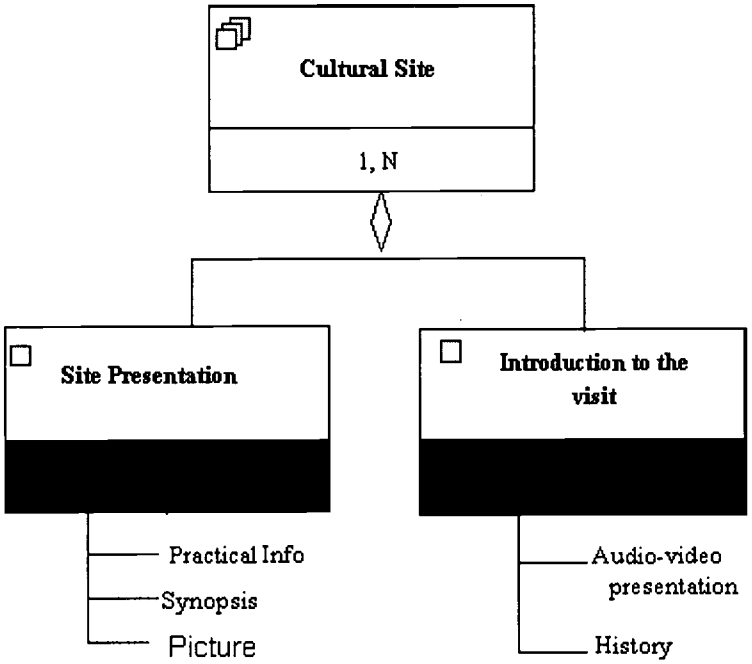


*Figure 4a: The content of Entity Type "Cultural Site"*

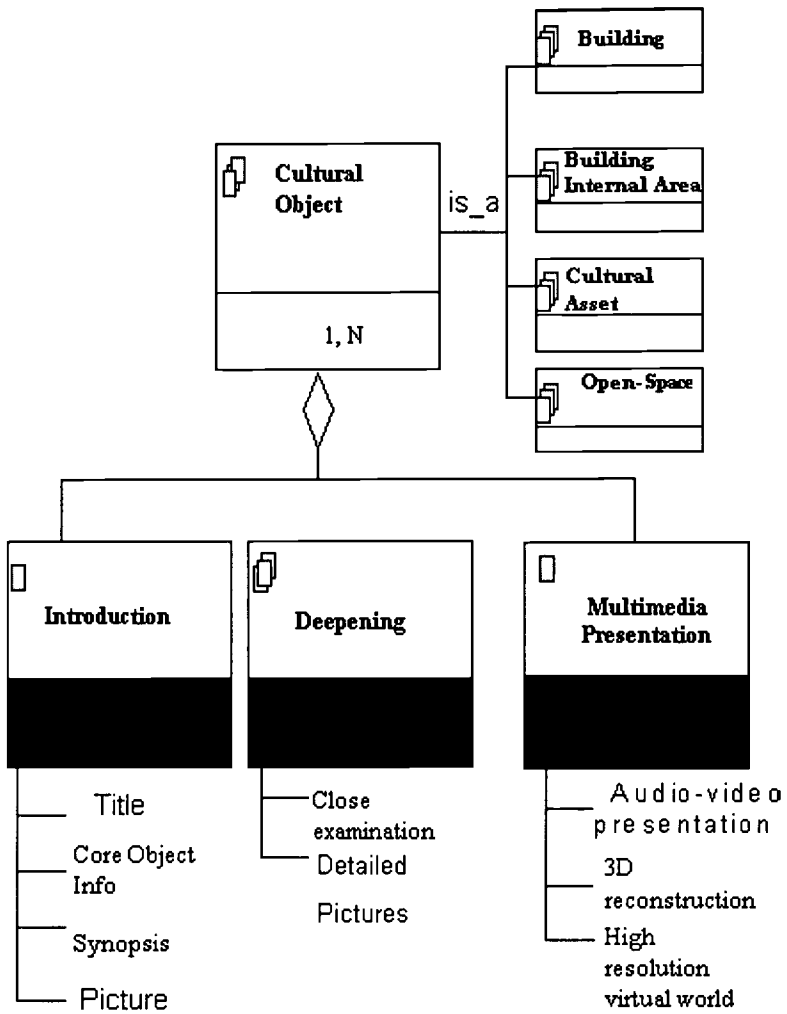


***Figure 4b: The content of Entity Type "Cultural Object"***

Hyperbase structures define the content "building blocks" of an MC application and classify them according to their intrinsic meaning and nature. Access Layer structure defines groups of hyperbase elements, called *Collections*, which provide complementary ways of organizing the contents. For example, a collection called "Highlights" is defined to group the most appealing Cultural Objects for a short visit of a cultural site. The goal of access layer structures is to drive the user towards the core application content that is represented by the hyperbase structures: During navigation, access layer structures are traversed before and in order to access the hyperbase.

According to the W2000 design model, a collection is described by its *members* and by a *center* . The center is an information element which describes the collection itself, e.g., by listing its members and by providing a short introduction to the collection subject. Collections can group hyperbase elements or other collections. In the latter case they are called *nested collections.*

In the MUSE application framework, the access layer structure includes the following collections:

- A collection for each entity type, which groups all entities of a given type ("Buildings", "Cultural Assets", Open Spaces", "Internal Spaces") . The center of these "By-type" collections includes a list of member descriptors (e.g. titles and miniaturized picture).

- A collection for each relevant theme, which groups the main entities relevant for a given topic. The center of a thematic collection includes a list of member descriptors (e.g. titles and miniaturized picture), and a short introduction to the collection theme.
- The collection "Highlights", which provides a selection of the most important elements for a short visit. Its center includes a short introduction to the visit, and the list of member descriptors which can be eventually visualized on a map showing where the various elements can be found on the cultural site.
- The collections "Geographic Tour", which groups "Cultural Objects" according to spatial criteria (e.g., in a large archeological site like Pompei, all monuments and assets "around the forum")
- The Collection "Album", which is created by the user while he or she is using the application as discussed in the memories channel.
- The nested collection "Index of Themes", which groups all the thematic collections
- The nested collection "Index of Geographic Tours", which groups all the collections' "Geographic tours"
- The nested collection "Index of Topics", which groups all the "By-Type" collections

### *MUSE Navigation Design*

The MUSE Information Schema only describe *which contents and groupings* are relevant for all possible channels of an MC cultural application. It does not provide any specification of which contents are navigated during the use of the application in a specific channel, nor how they are navigated. These aspects are described by the MUSE Navigation Schemas.

Each channel-specific navigation schema specifies a set of *node types* and *link types* . A node is an atomic unit of navigation and presentation in a specific channel. This means that navigation links connect nodes (and not node portions), and the effect of navigation is to activate an entire node: all content items in this node are loaded and made available on the page together (possibly together with other nodes).

During navigation design, the designer should

1. select the information elements in the information schema that are made available in each specific channel;
2. group them into nodes;
3. define the links among nodes that allow the user to exploit the semantic associations and the collections defined in the information schema.

The following diagrams show some portions of the MUSE framework navigation schemas for the different MUSE channels. Fat black arrows identify the entry to some sets of nodes and links; checkmarks identify "landmark" nodes, which can be reached from any node in the hyperspace.

These diagrams shows that the two on-site channels focus on multimedia content which has a more direct, emotional impact on the user and is a more appealing complement to what the user can experience on site. When the user employs these devices, multimedia contents are the first ones to be presented - audio-visual content for the mobile channel, and high resolution, interactive 3D content for the on-site stationary channel. Further (mainly based on static media) deepening can be retrieved on demand. In contrast, the navigation flow on the conventional Web channel starts from text and pictures, and proceeds to multimedia if the user wants further deepening.

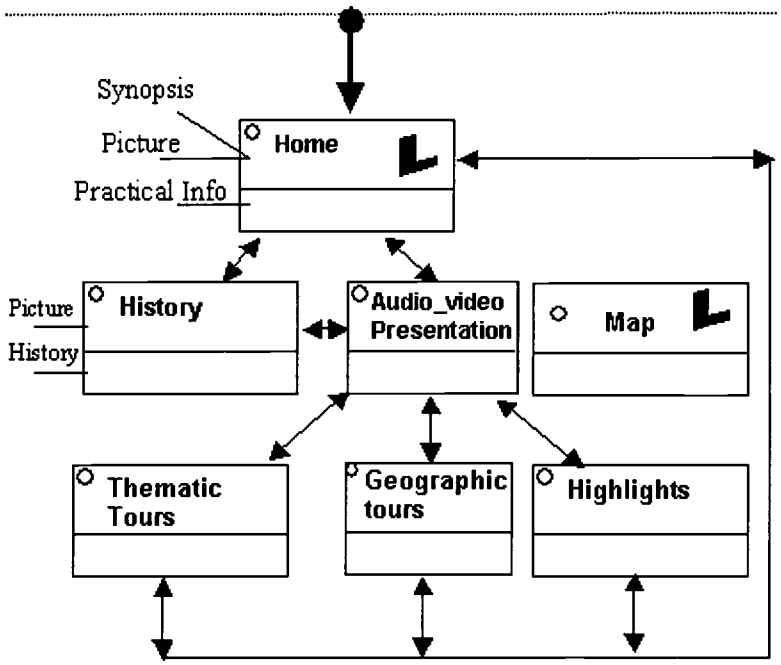


***Figure 5a: Conventional Web Navigation - High-level Access Structures Navigation Schema***

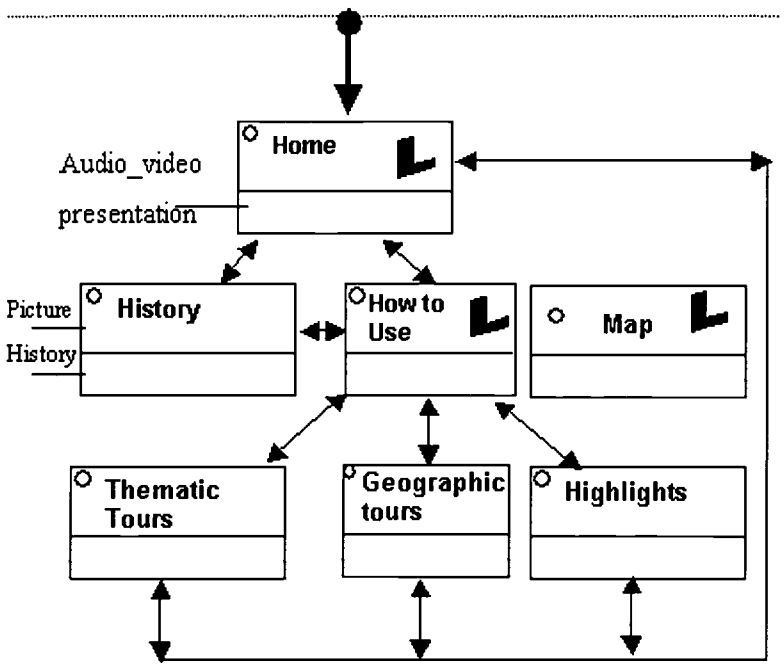


***Figure 5b: On-site Mobile Channel - High-level Access Structures Navigation Schema***

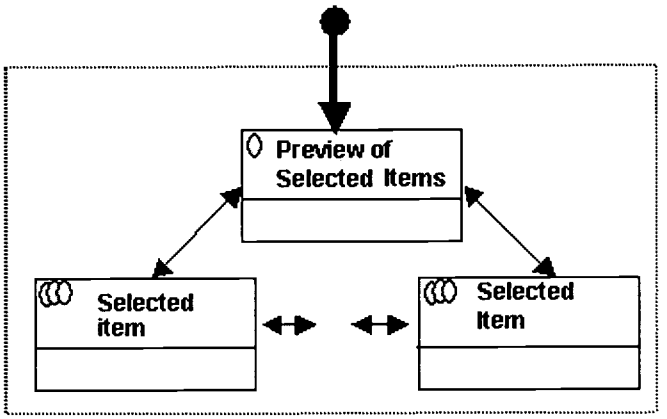


*Figure 5c: Memories Channel - Navigation Schema*

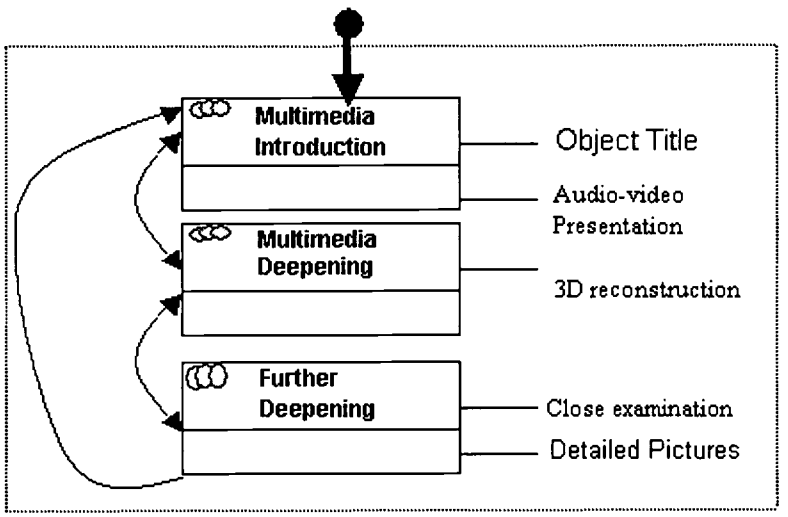


*Figure 6a: On-site Mobile Channel — Cultural Object Navigation Schema*

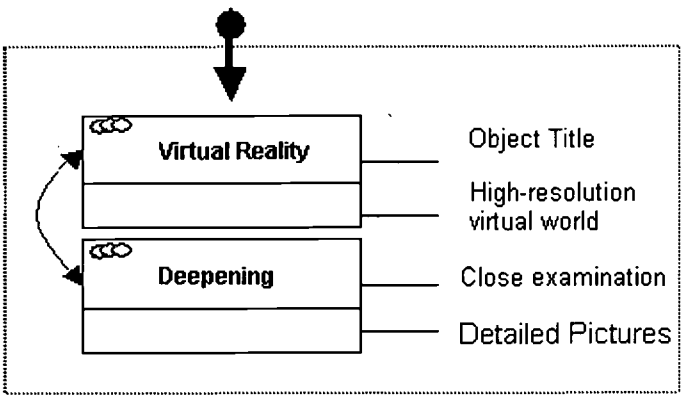


*Figure 6b: On-site Stationary Channel — Cultural Object Navigation Schema*

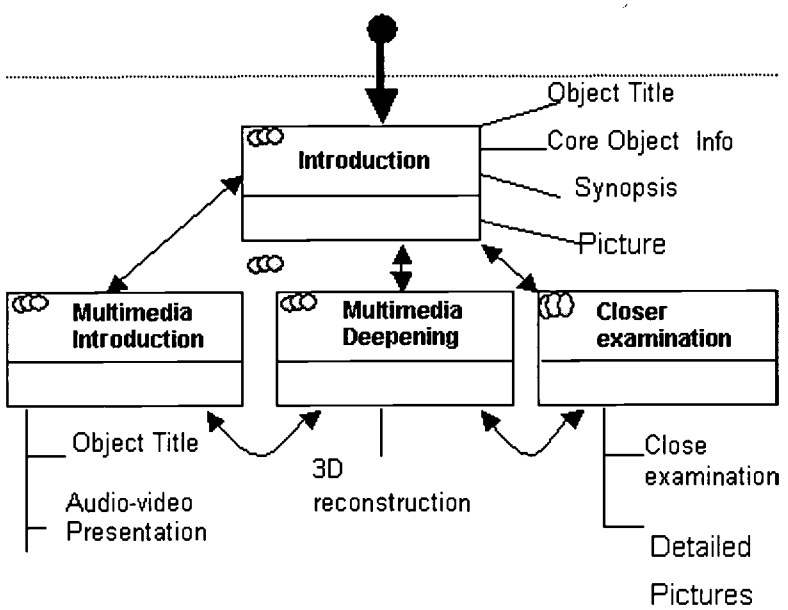


***Figure 6c: Conventional Web Channel — Cultural Object Navigation Schema***

### *Muse Interaction Design*

Interaction design defines how the users interact with the MC application in each different channel, and how the application behaves in response to user actions.

#### Interaction in the conventional Web channel

The interaction mode is *free browsing* , i.e., standard hypertextual navigation: user can activate any link defined in the navigation model, and the effect of link execution is to load and display the link destination node.

#### Interaction on the on-site mobile channel

MUSE supports three interaction modes: *free browsing*, *context-aware navigation,* and *memories-building.* Context aware navigation is the default, but the user can change interaction mode at any point during a session of use.

#### Free browsing

This is standard hypertextual navigation as defined for the conventional Web channel. The only difference is that the on-site mobile channel provides different content and navigation structures, as defined by the corresponding navigation schema. In free browsing, context-awareness capabilities of the device are off. What is presented on the device is totally under the user control and is not affected by his/her current position in the physical space.

#### Context-aware navigation

This interaction mode exploits the device capability of detecting the current geographical position of the user and of delivering context-dependent content. MUSE provides different forms of context-aware navigation which offer different levels of visibility of the content and different levels of control by the user:

**Pure context-aware navigation**

In this interaction mode, the application has the complete control. When the user reaches a place or an object in the tourist site associated to some audio-visual content, the device automatically delivers this information on the mobile device. It plays until its end, or until the user reaches a different position which is associated to a different content. The users are largely passive. They can only control the state of multimedia elements (e.g., suspending or resuming or stopping the audio-visual presentation) or switch to a different interaction mode.

**Guided context-aware navigation**

This interaction mode enhances pure context aware navigation with the notion of guided navigation. The application guides the users along a path on the site they are visiting, pinpointing the relevant objects there. These objects correspond to those grouped by the collections "Highlights", "Thematic Tours", "Geographical Tours". At any point of time, what is presented to the user is calculated by the application on the basis of both the user current position in the physical space and the currently active guided tour. The interaction metaphor is the vehicle satellite navigator: after the users have selected a tour of interest, the application highlights the path on an active map and shows the users' current positions. If the users reach a place or an object, either inside or outside the current path, which is associated to some audio-visual content, the device automatically loads and (dis)plays it. The application can detect any user deviation from the current tour path; wandering causes a sound warning and the display on the map of the shortest path to reach the currently active tour. In guided context-aware mode, the users are slightly less passive than in pure context-aware navigation. Beside controlling the state of multimedia elements (e.g., suspending or resuming or stopping the audio-visual presentation) and changing interaction mode, the users can ask for the map display, can change tour by returning to the tour selection menu, or can continue the exploration on a different tour. The latter possibility is available only when the users reach a place at the intersection of different tours. These tours are shown on the map and the users can choose any of them (the selection becomes the currently active tour).

### *Integrated context-aware navigation.*

It merges the capabilities of free navigation and context aware navigation. After the application displays some content in response to a change of user position or orientation, the users are allowed to explore related content and to browse as they can do in free navigation mode. Since the context-aware capability remains on, any physical movement of the users may cause a position-dependent replacement of the content which is currently active on the device.

**Memories-building mode**

The memories-building mode provides the users with the operations needed to manipulate the content for the CD memories channel. At any point during the use of the mobile device, the users can execute any of the following operations:

- mark the currently displayed screen or take a picture of what they are looking at (using a digital camera incorporated in Whyere). The effect of these operation is to update the memory album
- view and navigate the memories album
- modify the memory album by deleting some selected items, re-ordering them, or including comments and annotations

Any of the above operations activates the memories-building mode and suspends the current interaction mode, which can be resumed after the completion of the operations.

**On-site stationary channel**

In any place where the on-site stationary channel (a large screen controlled by a powerful graphic station) is available, users can switch to this channel and interact with it. What appears on the large screen is not a zoomed display of what is presented on the mobile device, but complementary high-resolution, interactive multimedia content which is related to the place where the stationary channel is located. Using the on-site mobile device as a control and pointing device, users can operate on the state of multimedia elements, explore the 3D space, and traverse navigational links. The content space available for navigation is a view of the whole content designed for the on-line stationary channel, filtered according to the position of the stationary device.

### *Muse Presentation Design*

The goal of presentation design is to define how contents and interaction elements are displayed on the different delivery channels. This activity comprises two sub-tasks: defining, for each channel, the *conceptual presentation* and the *concrete presentation* . Conceptual presentation defines *which* content and interaction elements are displayed simultaneously on the screen. Concrete presentation defines their lay-out graphical properties - space allocation, colors, typographical properties, images frames, etc..

The MUSE presentation framework mainly addresses conceptual presentation which can be reasonably shared and reused by a large number of applications in the tourism domain. Concrete presentation is largely "application specific" and may take into account the general visual communication strategy of the client institution.

The MUSE conceptual presentation framework comprises a presentation schema for each different channel. Each presentation schema defines a set of *conceptual page templates* which specify the types of nodes and links that must appear on the screen of a specific channel (Strothotte 1998). For lack of space, in this section we will focus on the presentation design for the on-site mobile channel only.

Given the limited size of the mobile device screen Whyre (6.4 inches), each page template includes a single node type and a placeholder for all link types outgoing from that node type in the navigation schema of the on-site mobile channel. Another constraining aspects of the on-site mobile channel which has an impact on presentation design is the lack of any pointing device, like a mouse, a trackball or a hand touch, to allow users to directly select any point on the screen. To interact with the system (e.g., to activate a link), the users must rely upon *physical buttons* available on the screen plastic frame. This factor introduces some intriguing requirements for concrete presentation: it forces the presentation designer to have a limited number of interaction elements on the screen and to place them in a position which allows users to "select" them using physical buttons only.

The Whyre "keyboard" includes four hardwired buttons and six "multimodal" buttons. The hardwired keys support the following functions: *undo, SOS (help), orientation, take a picture* .

The multimodal buttons are symmetrically located on both sides of the device. In this way, they are easily reachable by the users. As shown in the following pictures, each interaction element is placed on the screen close to a multimodal button which can be used to select that interaction element. Pressing the button executes the closest interaction element. The semantics of a multimodal button (which is the reason for this name) depends on the semantics of the closest interaction element.

The first screen displayed in figure 7 is the entry point ("home") of the Pompei application on the on-site mobile channel. The following interaction choices are available:

- Get an introductory presentation on the site
- Show the site map
- Select a navigation mode (one key for each mode)
- Get help on the current screen

By pressing the closest button to the "Site map label" (see left image — last button on the right side), the map of the archeological site, organized in different "insulas", is displayed (see right image).

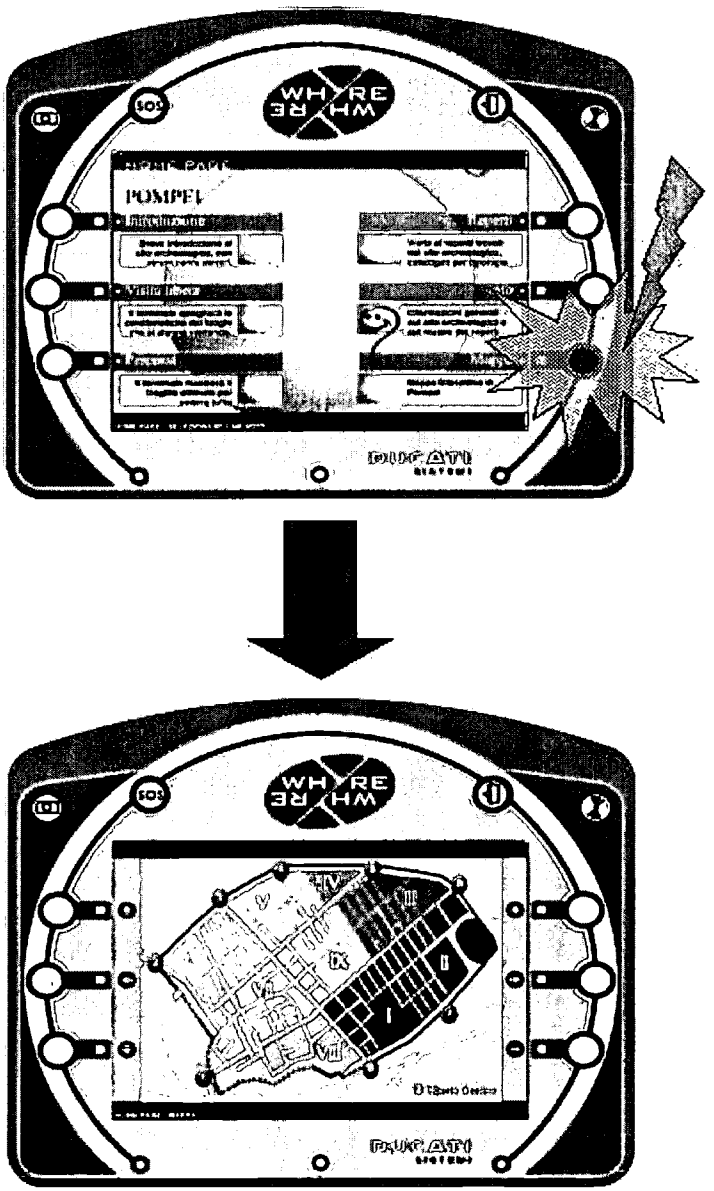


***Figure 7: From the virtual entrance of Pompei to the site map***

In Figure 8 we see the effects of pressing a different button (the mid right button in the left image). We suppose that the users are in the integrated context aware mode. When they enter *La Casa del Centenario* in Pompei, their position is detected by the application: an image showing the house atrium automatically appears (see left side image), and a sound comment starts. The users can choose a deeper description by pressing the mid left button, and can virtually enter the *atrium* as it was before the 79 A.C. Vesuvius eruption, according to the

reconstruction validated by the archaeologists (see the right side image) (Scagliarini, 2001).

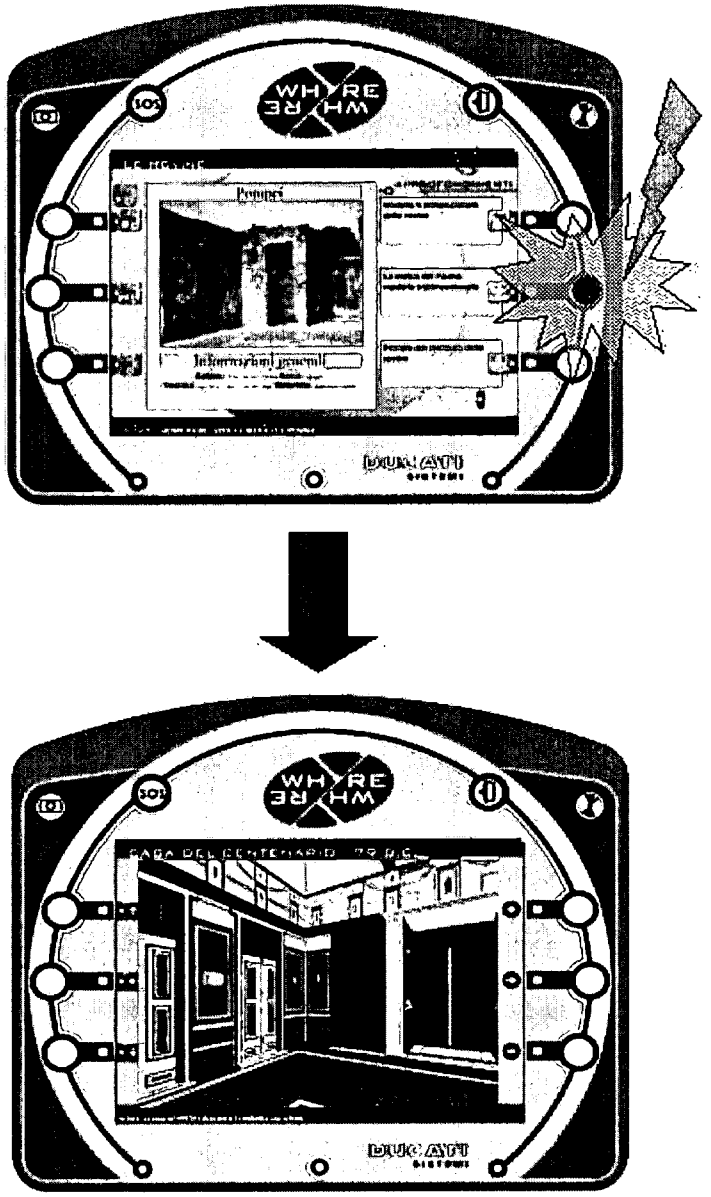


***Fig.8: Entering a virtual reconstruction of La Casa del Centenario while navigating in integrated context-aware mode***

Figure 9 provides an example of the multimodal behaviour of buttons and of guided context aware navigation. After the orientation hardwired button is pressed, the application displays the site map showing the visitors' position and orientation, as well as the list of the reachable places belonging to the current guided tour (left side image). If a desired place is selected by pressing the mid right button (see left image) the application displays a zoomed map (see right image), showing the area included between the visitors' current location and the selected section, and the recommended connecting path.

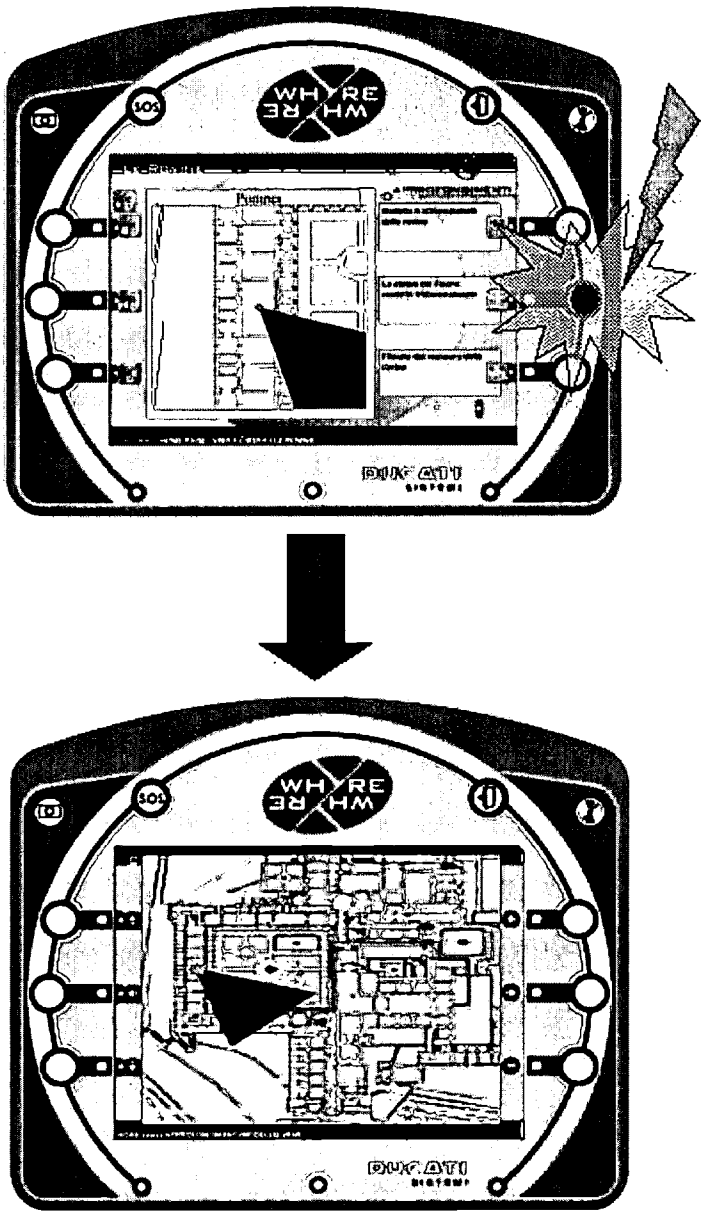


***Fig 9: Displaying a guided tour path in guided context aware mode***

**The Muse Interface Implementation Framework**

The Muse interface implementation framework is called *MuseXP*. It consists of a pair of interface definition tools (an editor and a compiler) coupled to an interface-independent execution engine (called *MuseXP Runner*). With *MuseXP*, MC systems interface design is approached at a higher level of abstraction with respect to standard Web applications. *MuseXP* targets are channel-specific interfaces in the cultural heritage domain. The framework is designed to be reconfigurable in order to be reused in other domains, should the opportunity arise.

Within *MuseXP* each channel-specific navigation and interaction design is mapped on to an XML file called *ch_interface.xml*. This file is a centralized interface control point easily handled by its creator and easily interpreted by the interface independent execution engine. XML (standing for *eXtensible Markup*

*Language*) was chosen for its extensibility: XML syntax was programmed with a number of tags finalized to the MuseMC system semantics. Interface presentation design is implemented filling in device-dependent HTML pages with the screen layouts and the graphic objects.

### *MuseXP Interface design workflow*

Given the conceptual design of a specific interface within a *MCMuse* system, how is the operational design carried out? How is the interface implemented on the target device?

In terms of the *MuseXP* architecture, these questions need to be read as follows: how are the *ch_interface.xml* file and the screen graphic layouts created? How are they processed on the target device by the *MuseXPRunner*?

Fig. 10 is a block diagram of the *MuseXP* system currently under development within the MUSE project.

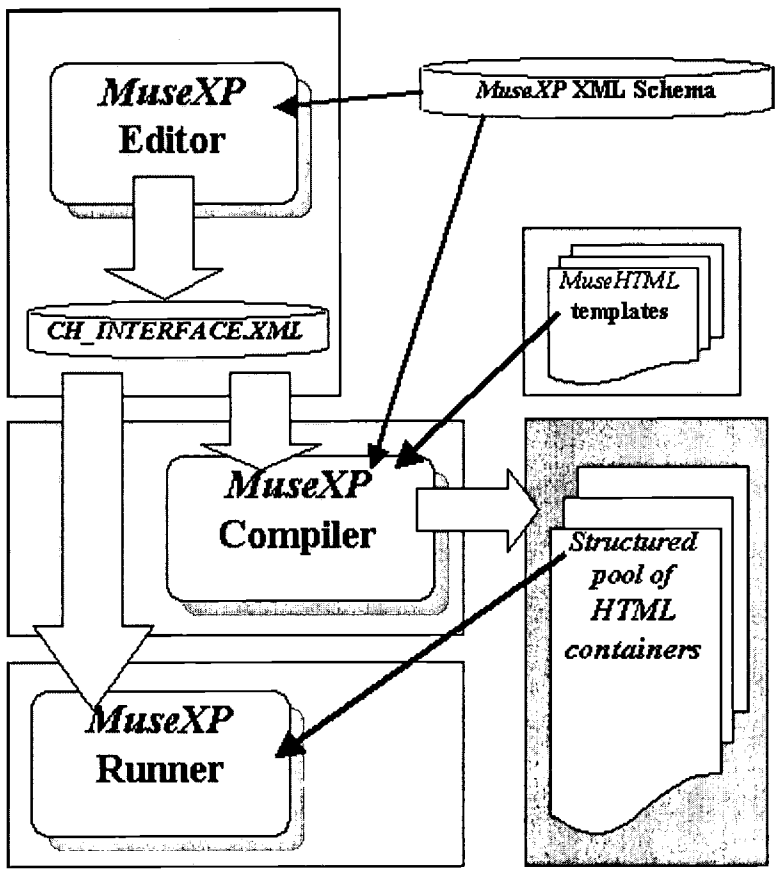


***Figure 10: Overview of MuseXP workflow***

Within the envisaged framework, the interface operational design is split into three steps:

1. Navigation and interaction specification
2. HTML containers generation
3. Interface graphic design

The first activity is expected to be carried out using a MUSE specific visual tool

called *MuseXP editor* currently under development. *Muse XP editor* maps the interface navigation and interaction designs into the text file *ch_interface.xml.* Since the editor is not yet available, *ch_interface.xml* has to be generated using an uncommitted text editor.

The second step is carried out automatically by an off-line tool called *MuseXP Compiler.*

The *compiler* generates a tree of all HTML pages required to implement the interface. These pages are called "*HTML containers* " because at this stage they only include the qualified structure-definition objects (i.e. the references to the required viewers and their attributes), but they do not include any graphic and metric layout information for the page presentation. The "HTML containers" pool is the set of all presentation instances, hierarchically arranged as declared in *ch_interface.xml* . There are no hypertextual links among these "Web pages". Navigation within the pool will be performed by the execution engine according to *ch_interface.xml* specifications. Each page is externally controlled according to the DHTML extension *DOM (Document Object Model).* In this way the contents to be delivered may be specified at run time.

The third step consists of the actual presentation implementation. During this step the graphic objects used throughout the interface are created and the *HTML containers* are filled up with the missing presentation items.

This activity can be carried out by Web interface designers or persons with similar skills, using standard graphics tools as well as HTML visual editors.

Once the HTML containers are graphically filled, they are saved on the target device with the *ch_interface.xml:* together, they specify the run-time rules-of-behaviour to *MuseXPRunner,* the execution engine that implements the designed interface.

The proposed framework has many advantages with respect to interface specific implementations:

- run time specification of both contents and page layout make a channel-specific interface reusable in many applications of the same domain
- an interface independent execution engine cancels programming from the list of design and implementation activities
- Separating structure from graphics is an additional answer to the complexity handling problem: it reduces the skills required for the most time-consuming tasks and it makes for easier design management (graphic layout, for example, can be modified without having to recompile the interface structure)

### *Conclusions*

State-of-the-art Web technologies offer the opportunity to deliver interactive multimedia content on-demand. Similarly, hardware technologies offer a wider and wider variety of devices, providing ergonomy and interaction modes much better customized to the users' context and location than a standard PC. These devices range from wireless connected mobile terminals exhibiting small and bright screens to large plasma display providing nearly immersive fruition in dedicated areas. The combination of these technologies within the same system offers the potential to provide ubiquitous services originating from the same information base and properly optimized to the fruition conditions. The user interfaces in such multichannel systems are different from standard Web clients. Web technology may be used, but the application design should be approached at a higher level of abstraction. Within the project MUSE this multichannel paradigm is currently under investigation. A framework supporting the design and implementation of heterogeneous interfaces in multichannel environments was developed, and its use in cultural tourism applications has been described in this paper. The framework was used to implement mobile and stationary on-site and

off-site channels for three MC systems addressing three cultural tourism "case studies": the *insula del Centenario* in the archaeological area of Pompei, the *History of Science Museum* in Florence, and a section of la *Certosa di San Martin,o* the outstanding historical Museum of the city of Napoli. The MUSE system will be demonstrated at these locations later this year, as soon as Whyre, the interactive multimedia terminal specially designed for on-site fruition in cultural heritage establishments, is ready.


### *Acknowledgements*

MUSE is funded by the Italian government (MIUR, the Italian Ministry for University and Research), within the framework of the National Research Program on Cultural Heritage Parnaso. The project is conducted by Ducati Sistemi S.p.A. in cooperation with Cineca, Politecnico di Milano, Sinet and the University of Bologna. Content providers as well as hosts for extended system concept verification are *Istituto e Museo di Storia della Scienza* (Florence), *Soprintendenza Speciale del Polo Museale di Napoli* and *Soprintendenza Archeologica di Pompei*. The authors wish to thank Prof. Pietro Giovanni Guzzo for establishing the connection between MUSE and the project "Pompei - Insula del Centenario"; Prof. Nicola Spinosa and his wonderful team for their continuous and irreplaceable support; Prof. Paolo Galluzzi for his conceptual contribution to the fruition model; Prof Francesca Bocchi for managing the training program associated to the research project; Ravi Nagaraj and Wen-Hann Wang with Intel Corporation for fruitful support in the design of Whyre hardware; Prof. Daniela Scagliarini and Antonella Guidazzoli with their teams for the development of the 3-D models of the *Casa del Centenario*.


### *References*


Abowd, G. D. (1999). Software Engineering Issues for Ubiquitous Computing. *International Conference on Software Engineering (ICSE)* , Los Angeles.

Baresi L., F. Garzotto, P. Paolini, (October, 2000). From Web Sites to Web Applications: New Issues for Conceptual Modeling. *WWW Conceptual Modeling Conference,* Proceedings, Salt Lake City.

Baresi L., F. Garzotto, P. Paolini, (January, 2001). Extending UML for Modeling Web Applications. In the *34th Hawaii International Conference on System Sciences* , P,roceedings Maui (USA).

Bocchi F. et al. (August, 1999). The 4D Virtual Museum of the City of Bologna, Italy, *Siggraph 1999* , Los Angeles.

Booch G., I. Jacobson and J. Rumbaugh (1998). *The Unified Modeling Language User Guide*. The Addison-Wesley Object Technology Series.

Conallen J. (Oct. 1999). Modeling Web Application Architectures with UML. *Communications of the ACM* , 42:10, 63-70.

Fayad M.E. and D.C. Schmidt, (Oct. 1997). Object Oriented Application Frameworks in *Communications of the ACM* , vol. 40, n. 10.

Garzotto F., P. Paolini, and D. Schwabe, (January, 1993). HDM - A Model-Based Approach to Hypertext Application Design. *ACM Transactions on Information Systems* , Vol. 11, No. 1.

Garzotto F., P. Paolini, L.Mainetti, (Aug. 1995). Hypermedia Design, Analysis, and Evaluation Issues. *In Communications of the ACM* , Vol. 38, N. 8.

Garzotto F., P. Paolini, D. Bolchini, and S. Valenti, (15 Nov. 1999). "Modeling by patterns" of Web Applications, in *Proc. Of the International workshop on the World-Wide Web and Conceptual Modelling* , WWWCM'99, Paris, 293-306.

Guidazzoli A. et al., (July, 2000). Tailored Virtual Tours in Cultural Heritage Worlds *Proceedings Siggraph 2000* , New Orleans.

Johnson R.E., (Oct. 1997). "Frameworks= Components + Patterns", in *Communications of the ACM* , vol. 40, n. 10.

Oppermann R., and M. Specht, (September, 1999). A Nomadic Information System for Adaptive Exhibition Guidance, *Proc. of the International Conference on Hypermedia and Interactivity in Museums (ICHIM)* , D. Bearman and J. Trant (eds.), Washington.

Ozen, B. T., Kilic, O., Altinel, M., Dogac, A., (May, 2001). Highly Personalized Information Delivery to Mobile Clients, *Proc. of 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '01)* , Santa Barbara, USA.

Perkowitz, M. and O. Etzioni, (May, 1999). Towards Adaptive Web Sites: Conceptual Framework and Case Study, *Proc. of the The Eighth International World Wide Web Conference, (WWW8)* , Toronto, Canada.

Salmon Cinotti, T. et al., (Milano, 2001). MUSE an Integrated System for Mobile Fruition and Site Management. In *ICHIM01* (*International Cultural Heritage Informatics Meetings* ; Milano, Politecnico, Sep 3-7 2001), Proceedings, Vol. I, 609-621.

Schmid, H.A., (Oct. 1997). "Systematic Framework Design", in *Communications of the ACM* , vol. 40, n. 10.

Scagliarini, D. et al., (November, 2001). Exciting understanding in Pompeii through on-site parallel interaction with dual time virtual models, *VAST01* , Glyfada (Greece).

Schwabe, D., G. Rossi, L. Emeraldo, F. Lyardet, (1999). Web Design Frameworks: An approach to inprove reuse in Web Applications. *Proceedings WWW99 Web Engineering Workshop* , Springer Verlag.

Strothotte, T., (1998). Computational Visualization - Graphics, abstraction and interactivity, Springer-Verlag.

Weiser, M., (July, 1993). "Some computer science issues in ubiquitous computing", *CACM* , Vol. 36, No. 7.

*U.S. Department of Education*
*Office of Educational Research and Improvement (OERI)*
*National Library of Education (NLE)*
*Educational Resources Information Center (ERIC)*

# NOTICE

## Reproduction Basis

[X] This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

[ ] This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").