IST-2002- 507382

EPOCH

Excellence in Processing Open Cultural Heritage

Network of Excellence

Information Society Technologies

D4.2.2: Report on Evaluation and Design Methodologies

Due date of deliverable: 1 April 2005
Actual submission date: 31 March 2005

Start date of project: 15 March 2004
Duration: 4 Years

PIN

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

1. INTRODUCTION

Web applications are essential to improve access for the citizen to Cultural Heritage (CH), to enhance education and tourism communication, and to contribute to the development of the new digital content and service industries. The web has promoted an increasing proliferation of on-line cultural applications. Almost any cultural institution or cultural initiative today wants to be on the web, and to promote their online presence, to disseminate its content, and its activities through the Internet. The web is now perceived as an "essential" mean of communication, given that online presence is a necessary counterpart of the presence of an event, an institution or an initiative in the real world.

However, CH institutions are poorly supported in designing and managing their online presence (by means of a variety of different channels, such as the traditional web, palm-held devices, 3G smartphones) in a coherent and effective fashion. On one hand, cultural institutions of medium/small scale (sometimes or frequently in Italy, but also in most of new accession countries) cannot afford to pay large and structured IT companies to do the job. On the other hand, large institutions often lack internal professionals with technological expertise, so they aren't able to discuss, to compare, to judge different methodologies and solutions in the ICT field.

As a consequence, the only possibility of CH institutions is to rely on house-made approaches (sometimes complemented by a small portion of actual outsourcing), where interactive online applications are developed through a "sketch & code" approach, thus comprising the overall quality of the online service. The lack of structured methods and good design processes may risk in failing to deliver quality applications, thus wasting precious resources for such crucial communication activities.

In this context, **methodological support** is strongly needed by CH institutions. In particular, methods covering the entire spectrum of the CH lifecycle activities are more and more demanded. Key activities include: requirements analysis, design, content development, implementation, evaluation, enhancement, management, promotion.
It is in fact increasingly recognized that a proper management of the lifecycle is key to the success and to the quality of online CH applications.

Namely, the **methods supporting design and evaluation** are among the crucial aspects to address and are the focus of this report. Effectively tackling design (crafting the user experience and the application) and evaluation (assessing the correspondence between the applications and the goals of each stakeholder) need proven methods and may lead to successful and quality applications.
Both in the professionals and the academic arena, a variety of approaches and methods exists, that can support somehow the design and evaluation activities for CH applications. Both in the academic and in the professional arena, the **methods available** represent a confusing scenario, mainly for the fact that they are highly **heterogeneous** for a number of reasons:

a) They are based on different assumptions about the users of the methods (developers, designers, project managers, other stakeholders) and their needs. What should designers need to effectively design a web application in the CH domain? Each method tries to answer to this question, and the answers are very different in nature.

Some methods assume that the "users" of the methods have a computer background. Some others consider also other types of users, such as content-experts, communication managers, interaction designers, ecc. As a consequence, some methods that fit for a community of users (but without an explicit intention) are not at all usable by other groups of developers.

b) They propose different **"visions"** about the role of design and evaluation in the lifecycle. Design and evaluations have different meaning in different communities of practice.

c) They are based on different **concepts.** Methods are coherent set of concepts that designers can user to carry out a given activity in the lifecycle. Some methods are based on formal concepts (common of software engineering and formal methods). Others are based of poorly-structured and volatile activities of design. Finally, proposals exist that blend a systematic approach with a light-weight one, offering a "semi-formal" method.

d) they provide different and heterogeneous **notations.** Tools for documenting the design and evaluation process and for specifying the results of these activities greatly vary in nature. Even here, there are extreme approaches: from graphical notations based on simple box-arrows diagrams to math formulas (especially for design).

e) they offer **tool support**, which is often proprietary, not compatible with other commercial tools and difficult to reuse and disseminate. When available, tool support is based on technology dated when the method was created. Moreover, tools available are rarely compatible or easy to be integrated with commercial tools available on the market.

f) Support and guidelines on the kind and size of **documentation** to produce for the design and evaluation activities is confusing and not at all uniform across the different communities of practice. It happens that the CH institution either wastes time and resources in filling out documents (without caring about the quality of the final application) or no documentation for the decisions taken along the process is kept.

In other words, current methods for coping with design and evaluation issues in CH are not at all standardized. There is a **lack of a shared standard** (even "de facto") in the design community that could facilitate the sharing of experiences, the promotion of case studies, and the dissemination of best practices among the communities of interest.

Lightweight design processes and usability are being recognized, more and more, as relevant for all design and evaluation methodologies, and for the design of interactive applications in particular. Different factors are being implied here:

- It must be **easy to teach** the methodology (and the design model) to anyone (from students to practitioners). Professionals, especially, do not have time and resources to invest for learning new methodologies; one of the success factors of "Entity Relationship" (probably the most successful design model, ever) stems from the fact that it was very easy to transmit its basic concepts, both in academia and professional environment.

- It must be possible to use the method for **brainstorming about design and evaluation**, i.e. for generating and discussing ideas among developers, with stakeholders, and with potential users). It is of little use to have a design model capable of representing only fully developed solutions.

- It must require **little time** to write down design and evaluation ideas: developers do not like to spend too many resources in preliminary activities.

- It must be possible to move, smoothly, from a **general design (and evaluation), to more detailed design (and evaluation)**, without need for excessive reworking and without need for completeness; in other words, even an incomplete design document must be useful and understandable.

The complexity and the "richness" of the design model is not what we are aiming for. Simplicity and "usability" of the design model itself, and of the corresponding design methodology, is our goal.

2. SURVEY METHODOLOGY

This report has three main source of information: the literature, the direct experience and the first results of a specific survey on the CH domain. Both POLIMI and UNISI are very active in the research community; our experts in each sector (design, usability, accessibility) are constantly up-to-date about and contribute effectively in the research trends of these domains.

Therefore, this report presents all the last enhancements and the main development directions coming out from the scientific literature. POLIMI and UNISI are also active in consultancy for institutions and companies involved in ITC projects in the CH domain; the information in this report is also influenced by these experiences in the "real world". Finally, some of our comments in this report are influenced by the very preliminary results of a questionnaire-based survey that POLIMI and UNISI are carrying out in the context of EPOCH WP 4.2 and IBC has reviewed.

The objective of this research is twofold. On the one hand, to investigate the current usage of design and usability evaluation methodologies, by identifying the most common methodologies and practices currently adopted by cultural institutions conceiving and evaluating their digital applications. On the other hand, to investigate the design and usability methodology needs, i.e., the actual desiderata by design or evaluation methodologies users (e.g., application analysts, usability experts, application developers, designers).

3. STATE-OF-THE-ART ON DESIGN METHODOLOGIES

Over the last decade, a number of structured design models and methodologies have been proposed for designing the features of an interactive application at a proper level of abstraction. Especially in the field of web design, various sets of concepts, process guides and notation primitives have been developed, partly extending existing modeling approaches for hypermedia (and traditional hypertext applications), and partly introducing novel concepts for dealing with website features. Roughly speaking, these approaches specify the design of a Web application at the conceptual level, neglecting technological aspects and constraints. Besides technical differences in concept used and application domains which they refer to, current design methods share lots of common features. All of them are based upon an information-navigation paradigm to describe the user interaction, whilst they differ one from another in terms of proposed design primitives, modeling language and support tools. On the other hand, all these methods recognize the importance of the semantics as guidance for the application design and share the fundamental principle of separation of concerns. Following this principle, and adopting the W2000 terminology (UWA Consortium, 2001), the design of a Web application can be divided into four different dimensions:

- Information and Access Structures design: defining the basic conceptual information units (entities) as perceived by the user and the navigational infrastructure in terms of semantics associations (relating different entities exhibiting some semantics relationship) and access structures (item collections guiding users in reaching interesting information on the basis of their needs).
- Operations and Business Process design: defining operations (e.g. "add to shopping cart") and processes (e.g. "check-out2, "registration") within a Web application.
- Navigation design: defining the navigation network allowing users browse information and access structures and execute operations and processes. Main modeling primitives are nodes (user consumption units of information or input forms in case of operations and processes inputs) and links allowing users move among them.
- Presentation design: defining the page structure in terms of lay-out aspects and graphical elements and the page organization and navigation. Page navigation is mainly based upon the navigation network specified in the previous design but introduce so called landmark links allowing short-cut navigation among pages.

For each design dimension designers have to perform a limited set of decisions which contribute partially to the overall user experience. Design primitives, belonging to each design dimension, embody a semantics focusing on a part of the overall experience users have when they face a Web page or navigate among several ones.

Although the design methods and models share at a higher level a common goal, which is enabling to take and document design decisions before implementing the application, they have several differences, including their main application domains, the level of coverage of the design process, and the level of support provided at different stages (Woukeu et al., 2003).

3.1 HDM

HDM (Hypermedia Design Model) is an early E/R-based design model proposed by Garzotto et al. (Garzotto et al., 1993) to define the structure and navigation of large-scale and read-only hypermedia systems. HDM prescribes the definition of an application schema, which describes overall classes of information elements in terms of their common presentation characteristics, their internal organization structure, and the types of their mutual interconnections. The model is suitable for domains with a high level of organization, modularity and consistency. HDM is organized in the following parts:

- Information Design: the purpose is the identification of the relevant information to be handled by the application, and the provision of an overall organization of the information structures, independently from any specific intended usage. The main concept introduced is the distinction between the Hyperbase and the Access Layer. The Hyperbase is the place where most of the content is defined, and is the most stable part of the application. The main design notions for the Hyperbase are the following:
    - Entity: a virtual object of interest for the user, organized into components
    - Components: the constituents of an entity
    - Slots: the atomic elements of content, constituents of components
    - Semantic Associations: the connections among the parts of the Hyperbase, that provide the "infrastructure" for possible navigation
- Entities may exist as instances of Entity Types or as Single Entities (one-of-a-kind objects).
- The main design notions for the Access Layer are the following:
    - Collection: a set of objects (called "members" of the collection), grouped together in order to provide a meaningful access to information for the user
    - Collection center: a new piece of information (e.g. an index) allowing the user the make the best possible usage of a collection
    - Collection Topology: the inner organization of a collection, for allowing an optimal "reading" of it
    - Collection Filter: providing the possibility of searching, within a collection, members with specific properties
- Collections may exist as instances of Collection Types or as Single Collections (one-of-a-kind groupings).
- Navigation Design: the purpose is the organization of the information into pieces (nodes) oriented toward "user consumption"; also navigation paths are provided in order to allow user navigation, which is one of the most distinctive features of hypermedia applications; it is organized upon a limited set of design notions. The main concepts are the following:
    - Node: a set of slots specifically chosen in order to provide a "unit of consumption" for the users. Nodes may appear as instances of node types or as single nodes (one-of-a-kind units). Nodes, for the gretest parts, derive from Entities, or Sematic Associations or Collection centers.
    - Cluster: a set of nodes, which can be navigated across. Clusters of nodes are generated from Entities (Structural Clusters), from Semantic Associations (Association Clusters) or Collection (Collection Clusters).
    - Accessibility relationship: a connection among nodes that support navigations. Accessibility relationships "tie" together the nodes of the same cluster, supporting navigation within it.
    - Navigation Pattern: it describes the actual way navigation across nodes (within a cluster) is allowed. The model defines a set of navigation patterns (such as index,

guided tour and all the variants deriving from the combination of these basic patterns), which are recurrent strategies for organizing the user's navigation among information units, entities and collections.

– Publishing Design: the purpose is the organization of the information into "pages", which are the units of "physical consumption" for the user. The main notions being used are Page, Section, Publishing Units and Interaction Spot.
    - Page: is the "physical unit delivered to the user. A Page can be an instance of a Page Type, or a Single Page. A page is organized into Sections. The sections within the same page are semantic units, independent one from the others.
    - Section: a section is the semantic constituent of a page. It contains a single piece of information (roughly corresponding to a node) or a set of correlated pieces of information (roughly corresponding to a cluster). A sections consists of one or several publishing Units.
    - Publishing Unit: it is the atomic part of a section. It may derive from the navigation design ("content" publishing unit), or provide new content added for publishing purpose ("decorator" publishing unit) or provide interaction elements ("interaction" publishing units).
    - Interaction Spot: it provides a way to "operate" with the application. An interaction spot may provide a link to a different page (instance), a link within the same page (instance), an interaction with the content (e.g. zoom, play, start, stop, …), the activation of an operation (see below) or the control over a transaction execution (see corresponding document).
– Operations Design: the purpose is to provide the user with operations (other than standard hypermedia operations, such as navigation) that allow the user to invoke application dependent "functions" (such as "register", "submit", etc.). It provides the ingredients to add operational functionalities to hypermedia applications. The main design notions are the following:
    - Pre-conditions: the conditions that must be satisfied before the user can invoke the operation. Preconditions may involve the "hypermedia state" (i.e. the state of the objects defined by the hypermedia design), or the "application state" (i.e. the state of additional information handled by the application) or other system and environment states.
    - Post-conditions: the conditions that will be satisfied after the completion of the operation. Post-conditions may involve all the elements of preconditions and also "output", i.e. actions sending "messages" outside the strict environment of the application.

An extended and richer version of HDM, called W2000, has been defined to cope with the complexity of web information systems (UWA Consortium, 2001).

3.2 RMM

RMM (Relationship Management Methodology) (Isakowitz et al. 1995) is E/R-based, suitable for structured hypermedia applications and its design process consists of seven steps: entity-relationship design; slice design (grouping entity's attributes as node/presentation units called slice); navigational design (access methods: link, menus, index, guided

tour); protocol conversion design (converting design components into physical objects); user interface design (screen layouts); run-time behavior design and construction and testing.

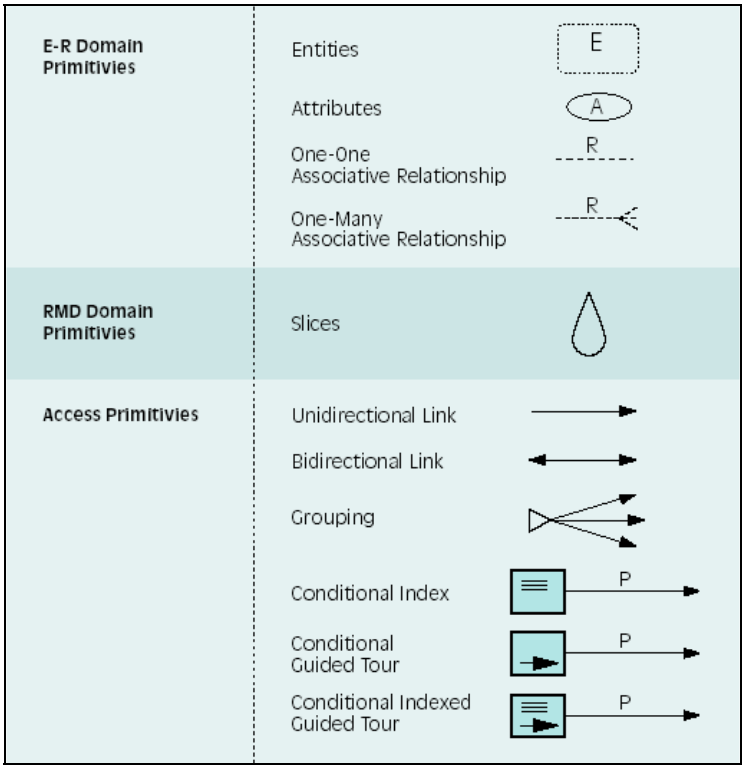

Fig. 1. The Relationship Management Data Model primitives

RMM is focused on the design phase and employs the Relationship Management Data Model (RMDM), which represents an integration of E-R and HDM models. RMDM provides a language for describing information objects and navigation modality in hypermedia applications. The language primitives are illustrated in Fig. 1. The RMDM primitives can be divided into E-R primitives, m-Slice domain primitives, and access primitives.

– E-R primitives are taken from the Entity-Relationship modeling, and describe how information is structured in the application domain. Entities and their attributes represent abstract or concrete objects and the relationship between entities are modeled by 1-1 or 1-n associative relationships;
– m-Slices are used to model small information units of a presentation. Indeed, m-Slices are defined by grouping attributes of a given entity of the E-R diagram or by grouping others m-Slices. The 'm' refers to Russian Matrjeska dolls, because m-Slices are nested in other m-Slices until the presentation unit is defined. This approach let the design of user interface be more modular and support reuse, because m-Slices can be employed several times in the presentation.
– access primitives allow us to model and to support navigation across different entities. An RMDM diagram describes how users will navigate across a hypermedia application. In Fig. 1 the access primitives are classified into unidirectional links; bidirectional links; grouping, conditional guided tour, conditional index, and conditional indexed guided tour. A guided tour is a linear path through a collection of elements allowing the user to move either forward or backward on the path. An index is a table of content to a list of entity instances, providing the access direct to all the elements of the list. A grouping is a mechanism serving as an

access point to other parts of the hypermedia document. For example, the initial screen of many applications contains a menu or a set of buttons that provide access to different functions or classes of information. It is also possible to combine index and guided tour to have the indexed guided tour. A predicated is associated to the links of the access primitives to indicate which instances of an entity are accessible through that primitive. For example, the predicate "(rank='associate')" indicates that only the entities whose rank is associate are involved in the guided tour, obtaining a conditional guided tour.

RMM describes completely the software development cycle, but focuses on the critical design phases. It does not address early and late lifecycle activities, such as project management, feasibility studies, requirement analyses, planning, evaluation and maintenance. The methodology foresees an iterative approach to the development since it encourages feedback between the various development steps.

Let us give a brief description of the eight steps forming RMM, some of which can be conducted in parallel. During the Entity-Relationship design the designer models the information domain and its relationship via an E-R diagram. This model is familiar to the designers, well documented and can be used for modeling the information dependencies in a variety of application domains. This kind of analysis helps to determine a view of the application domain independent from the presentation and to identify relationships across which navigation can be supported. Moreover, if the target application is a Web interface for an existing database, its E-R diagram can be reused.

The next step, the application diagrams design, consists in modeling the information at presentation level and focuses only on the presentation structure. Application diagrams show the content of each web page and the connections among them. During the m-Slice design (E-R+ diagram) the attributes of one or more entities that should be displayed simultaneously on the screen are modeled and details are hidden, such as elements of the user interface or details nested in other m-Slices. The methodology suggests how to create this kind of diagrams in detail.

Navigational design: this phase allows the designer to establish both how users will access to information and the paths for the hypermedia navigation. To this aim, designers specify menu-like structures using indices and guided tours. The lower level structure can be collected in higher level structure, obtaining a hierarchical access based on menu. At the end of this step all the access structures will be described by a RMDM diagram.

The remaining steps are not characteristics of this methodology and are briefly illustrated in the following. In the conversion protocol design phase the designer describes how abstract constructs have to be transformed into physical-level constructs; for example, it is possible to define how a slice is converted into an HTML Web page, or an index can be implemented by using an HTML form. The User-Interface Design and Construction step considers how the information structured in the previous phases has to be presented to the final user. For example, index aspect, button layout, etc. In this phase it is important to take into account that for many WWW applications it is crucial to exhibit a common look and feel; Run-time behavior design describes the functionality to be realized at run-time; for example, possible inclusion of search engines, dynamic generation of pages, backtracking that allows the user to go back to previously visited nodes; history lists, maintaining an ordered list of each visited node, etc. The methodology let the designer use both top-down and bottom-up approach. The development process results flexible and iterative and the quality of the final product is improved because it turns out to be structured, extensible, maintainable and reusable. Another appealing feature of the RMM methodology is the availability of a software tool, named RM-Case, supporting RMM design and development and producing the diagrams associated.

3.3 OOHDM

OOHDM (Object Oriented Hypermedia Design Model) (Rossi et al. 2001) is an OO-based design model, extending HDM, that allows the specification of hypermedia applications as navigational views over a conceptual model. It allows us to build large hypermedia applications using a mix of incremental and prototype-based process model. OOHDM supports the most part of development process, except the early phases of the lifecycle. It consists of four steps, namely conceptual design, navigational design, abstract interface design and implementation. Each step allows us to build an object-oriented model focused on a design aspect. These steps can be accomplished separately, obtaining a reusable and modular design.
Conceptual Design: An object model is used to provide a conceptual model of the application domain. This model refers to the usual object-oriented modeling principles, extended by concepts derived from HDM and similar to perspectives.
Navigational Design: During this phase the information described in the conceptual model is structured to define different views, a view for each user profile. To this aim, OOHDM defines a navigational model that takes into account the different user types and the actions they can perform. Navigational design is accomplished through the definition of two schemas: the navigational class schema, and the navigation context schema. The navigation class schema let the designer define classes having similarity with the conceptual classes, but with several differences. For example, in a fly-company Website, a node class "Fly" for a travel agent does not contain several attributes of the "Fly" conceptual class; e.g., crew name, etc.... as they represent information which would not be visible in the travel agent view. The navigation context schema allows the designer to define which information is reachable in a given node in a particular context. Thus, a navigational context is a set of nodes, links, context classes and other (nested) navigational contexts.
Abstract Interface Design: The application interface is defined through the abstract interface model. It is built by defining interface classes, i.e. objects visible to the user (e.g. a picture, a city map, etc.). They are an aggregation of primitive classes, for example buttons and text fields, and recursively of interface classes. In this phase the designer establishes how to activate the navigation and to realize the synchronization among multimedia interface objects. Here, the association between interface objects and navigational objects is created. Interface behavior is declared by specifying how to handle external and user-generated events and how the communication takes place between interface and navigational objects.
Implementation: This phase consists in mapping the interface objects into implementation objects, related to the implementation environment that can involve different platforms, such as, Hypercard, Toolbook, Director, HTML. The OOHDM methodology improves maintainability and reusability, thanks to both the separation of the design phases and the abstraction capabilities which are characteristics of the object-oriented design.

OOHDM extends the model HDM by including special purpose modeling primitives for both navigational and interface design. Moreover, it introduces the navigational contexts and offers a great relevance to the interface design. In fact, it models the man-machine interaction and considers the effect of each event generated by the user both on the interface and the navigational aspects. The design and generation of OOHDM-based read-only web sites is supported by a CASE tool called OOHDM-Web. Like HDM, the extensive use of design patterns of different kinds (concerning navigation, information structuring and presentation strategies) enables efficient documentation and reuse.

3.4 OO-H

Following a similar object-oriented approach to conceptual design, OO-H method (Object-Oriented-Hypermedia Method) (Gómez et al., 2001) proposes a sound design process for specifying the features of a web application independently from their fruition device. The OO-H method is a generic model providing designers with the semantics and notation necessary for developing Web-based interfaces and connecting them with previously existing application logic modules, thus facilitating applications migration. To achieve this goal, the authors based the OO-H method on the information reflected in a UML-compliant approach, known as the OOmethod. For the authors' purposes the OOmethod is an automated software production environment whose main constituents are:

- a set of views to capture the system structure (statics) and behavior (dynamics), and
- a model compiler that generates the data sources and the logic modules in the desired implementation environment.

The OO-H method extends these views with two new complementary diagrams. The navigational access diagram (NAD) defines a navigation view, and the abstract presentation diagram (APD) gathers the concepts related to presentation. Both the NAD and the APD capture the interface-related design information with the aid of a set of patterns, defined in an interface pattern catalog integrated in the OO-H method proposal. Following the OO-method philosophy, the OO-H method provides a model compiler that generates the Internet application front-end for the desired client platform and/or language (HTML, XML, and Wireless Markup Language, or WML). This extension provides a true three-tiered Internet solution.

The OO-H method includes the following set of notations, techniques, and tools that make up a sound approach to the Web product modeling phase: a design process, a pattern catalog, a NAD, an APD, and a CASE tool that automates the development of Web applications modeled with the OO-H method.

The design process defines the phases the designer has to cover for building a functional interface that fulfills the user requirements. The OO-H method design process departs from the domain information structure captured in a UML-compliant class diagram. From there, the method models personalized (1..N) different NAD instances, one for each user type. Each NAD instance reflects the information, services, and required navigation paths for the associated user's navigation requirements fulfillment. Once the NAD has been constructed, the model allows generating a default Web interface following a set of mapping steps. This automatic generation feature lets the designer shorten the time necessary to develop application prototypes. However, final implementations usually require a much higher level of sophistication, both from the visual and the usability point of view.

To improve the interface quality, the OO-H method introduces a second diagram—the APD, based on the concept of templates—and directly derives its default structure from the NAD. To help the designer refine this structure while maintaining its quality, the pattern catalog contains a set of constructs that effectively solve problems identified within Web environments. This approach facilitates the reuse of design experiences and the consistency among the different interface modules and among application interfaces. Once the APD is refined, the model allows generating a Web application front-end—either static or dynamic—for the desired environment,

such as HTML, WML, active server pages (ASPs), and JavaServer pages (JSPs). Again, the designer can define different (1..N) APDs for the same NAD, reflecting different ways of visualizing the same navigation requirements. This independence from final implementation issues proves necessary in an environment where new appliances and languages for Internet access emerge constantly.

The OO-H method provides also a pattern catalog and a hypermedia interface pattern language that can be seen as a partially ordered collection of related patterns that work together in the context of hypermedia interfaces and that helps capture the abstract interaction model between the user and the application.

3.5 EORM

The Enhanced Object-Relationship Model (EORM) (Lange, 1996) is defined as an iterative process concentrating on the enrichment of the object-oriented modeling by the representation of relations between objects (links) as objects. According to Lange, this has the following advantages: relations become semantically rich as they are extensible constructs, they can participate in other relations and they can be part of reusable libraries. This method proposes the construction of a prototype of the user interface at an early stage during design. The method is based on three frameworks: class, composition and GUI.

- The class framework consists of a reusable library of class definitions. To identify classes for an application EORM follows standard object-oriented techniques. EORM distinguishes two types of object-oriented relationships: generalization relationships and user-defined relationships. Whereas the first ones have predefined semantics associated to them; the second ones rely completely on the user specification.
- The composition framework consists of a reusable library of link class definitions that enable users to reuse already developed link classes and extend them when necessary by using inheritance. The semantics of the basic link classes is the following:
    - simpleLink: is the root link class that provides basic interlinking capabilities, including functions for creation, deletion and traversing.
    - navigationalLink: provides traversal mechanisms for hypermedia links, including storage of creation time and history information (backtrack). It inherits from simpleLink.
    - nodeToNode: is a link that inherits from navigationalLink providing an object-to-object hypermedia link functionality.
    - spanToNode: inherits from navigationalLink. It links the content of an object to another object.
    - structureLink: is a child of simpleLink and the root of the structural links. It is inserted after creation into the structural context.
    - setLink: is a structureLink that provides access to an object in an unordered collection of objects.
    - listLink: is a structureLink that provides access to an object in an ordered collection of objects.
- The last step of this method is the design of the GUI application using elements of the GUI Framework. It proposes to determine the windows of the domain and which presentation has to appear in each window, to obtain presentations from attributes and operations of classes and determine how functionality is assigned to window menus.

The ONTOS Studio tool was developed to support the hypermodelling process with EORM. It utilizes an interactive graphical user interface that generates a C++ implementation of the hypermodel. It is based on the ONTOS database.

3.6 WDSM

The Web Site Design Method (WDSM) (De Troyer et al., 1997) is a user-centered approach which starts from considering user classes and their requirements in terms of preferences and views. On this basis, the model proposes to design an information-intensive website in three main stages: User Modeling, Conceptual Design, Implementation Design and the actual Implementation. The User Modeling phase consists of two sub-phases: User Classification and User Class Description. The Conceptual Design phase also consists of two sub-phases: the Object Modeling and the Navigational Design.

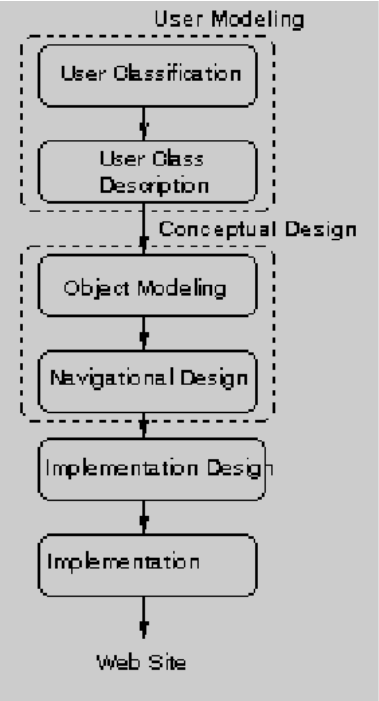

Fig. 2: Overview of the WSDM phases

- User Modeling. Users usually visit web sites with questions in mind. The web site should anticipate the user's questions and answer them. Therefore, the first phase of this method is concentrated on the potential users of the web site.
    - User Classification. In this step the designer identifies and classifies the audience of the web site. The activities and parties involved can be represented in a schema.
    - User Class Description. A user class is a subset of the all potential users who are similar in terms of their information requirements. Users from the same user class have the same information requirements.
- Conceptual Design.

- Object Modeling. Here the information requirements of the different user classes and their perspectives are formally described. This is done by building (conceptual) object models for the different user classes.
- Navigational Design. Here the designer describes how the different users can navigate through the web site. The model consists of a number of navigation tracks, one for each perspective. This is described in terms of components and links . We distinguish information components, navigation components and external components. Each navigation track consists of three layers:
  - The context layer. The navigation track starts with a navigation component with the same name as the perspective. This forms the top level of the navigation track.
  - The information layer. Each perspective becomes an information component or an external component in the navigation track. The choice between information component and external component is determined by the availability of information. For each relationship between perspectives a link is created.
  - The navigation layer. This layer connects the context layer and the information layer and has to be designed according to the needs of the users of the perspective.

– Implementation Design. In this step the designer essentially designs the 'look and feel' of the web site. The aim is to create a consistent, pleasing and efficient look and feel for the conceptual design made in the previous phase. It should be noted that the design of the implementation may depend on the chosen implementation environment. Limitations of the implementation language (such as HTML) may interfere with this process.
– Implementation. The last phase is the actual realization of the web site using the chosen implementation environment. For example, for an HTML implementation this means that the implementation model must be converted into a set of files containing HTML source code. This step can be largely automated using available tools and environments for assisting in HTML implementations. To improve maintainability, more and more web sites have their underlying information (or parts of it) stored in a database. Pages are generated (usually, but not necessarily, at runtime) by extracting the necessary information from the database. This information may be extracted from existing databases or a new database may be implemented for this purpose.

3.7 WebML

WebML (Web Modeling Language) is a high level, model-driven, and E/R-based (compatible with UML class diagrams) design approach allowing a conceptual specification and automatic implementation of data-intensive websites (Ceri et al, 2002). WebML enables designers to express the core features of a site at a high level, without committing to detailed architectural details. The approach proposes a structural model (data design), a hypertext model, a composition model, a navigation model, a presentation model and a personalization model. The model has also a CASE tool called WebRatio. WebML also supports an XML syntax, which instead can be fed to software generators for automatically producing the implementation of a Web site.
The specification of a site in WebML consists of four orthogonal perspectives:
– Structural Model: it expresses the data content of the site, in terms of the relevant entities and relationships. WebML does not propose yet another language for data modeling, but is

compatible with classical notations like the E/R model, the ODMG object-oriented model, and UML class diagrams. To cope with the requirement of expressing redundant and calculated information, the structural model also offers a simplified, OQL-like query language, by which it is possible to specify derived information.

- Hypertext Model: it describes one or more hypertexts that can be published in the site. Each different hypertext defines a so-called site view. Site view descriptions in turn consist of two sub-models.
    - Composition Model: it specifies which pages compose the hypertext, and which content units make up a page. Six types of content units can be used to compose pages: data, multi-data, index, filter, scroller and direct units. Data units are used to publish the information of a single object (e.g., a music album), whereas the remaining types of units represent alternative ways to browse a set of objects (e.g., the set of tracks of an album). Composition units are defined on top of the structure schema of the site; the designer dictates the underlying entity or relationship on which the content of each unit is based.
    - Navigation Model: it expresses how pages and content units are linked to form the hypertext. Links are either non-contextual, when they connect semantically independent pages (e.g., the page of an artist to the home page of the site), or contextual, when the content of the destination unit of the link depends on the content of the source unit.
- Presentation Model: it expresses the layout and graphic appearance of pages, independently of the output device and of the rendition language, by means of an abstract XML syntax. Presentation specifications are either page-specific or generic. In the former case they dictate the presentation of a specific page and include explicit references to page content (e.g., they dictate the layout and the graphic appearance of the title and cover data of albums); in the latter, they are based on predefined models independent of the specific content of the page and include references to generic content elements (for instance, they dictate the layout and graphic appearance of all attributes of a generic object included in the page).
- Personalization Model: users and user groups are explicitly modeled in the structure schema in the form of predefined entities called User and Group. The features of these entities can be used for storing group-specific or individual content, like shopping suggestions, list of favorites, and resources for graphic customization. Then, OQL-like declarative expressions can be added to the structure schema, which define derived content based on the profile data stored in the User and Group entities. This personalized content can be used both in the composition of units or in the definition of presentation specifications. Moreover, high-level business rules, written using a simple XML syntax, can be defined for reacting to site-related events, like user clicks and content updates. Business rules typically produce new user-related information (e.g., shopping histories) or update the site content (e.g., inserting new offers matching users' preferences). Queries and business rules provide two alternative paradigms (a declarative and a procedural one) for effectively expressing and managing personalization requirements.

In the ToriiSoft tool suite, WebML specifications are given as input to a code generator, which translates them into some concrete markup language (e.g. HTML or WML) for rendering the composition, navigation and presentation, and maps the abstract references to content elements inside pages into concrete data retrieval instructions in some server-side scripting language (e.g., JSP or ASP).

3.8 IDM

IDM (Bolchini & Paolini, 2004) is an Interactive Design Model specifically tailored for designing effectively interactive applications such as web sites, mobile applications, etc. This approach derives from the observation that we can consider a web experience as a dialogue between the user and the web site being used; the background research, moving from linguistic theories and practices, has led to the development of a design model based on dialogue primitives and characterized by a limited set of dialogic concepts used to shape the interaction between a user and the application. IDM is based on the concepts of "topic", "kind of topic" and "relevant relation". Beside the emphasis on dialogic interaction, IDM has additional distinctive features: it is lightweight (easy to learn and to teach), suitable for brainstorming at early stage during design (or during the shift from requirements to design), cost-effective (it requires little effort from designers) and modular (designers can take the part they wish, not being forced to "all or nothing"). IDM provides a simple notation to describe graphically the (conceptual and logical) structure of an interactive application; the notation is summarized in the example in Figure 3.
IDM overall organization is structured in three main steps: Conceptual Design, Logical Design and Page Design. The Conceptual design identifies the main subject of the conversation that the application can perform with the user and the main strategies used to start talking about a specific subject. The Conceptual design is an "absolute" view of the application, independently from the channel used. The Logical design introduces all the details that are strictly dependent from the channel, i.e. a web site, a mobile WAP application, an iTV application, etc. The Page design organize into pages the contents and the interaction capabilities described in the two first schemas; this part of the design do not deal with graphics or layout problems but design each page of the application describing the contents and the links that will be visualized.

- The Conceptual design (C-IDM) of an interactive application modeled with IDM, must convey all the necessary "dialogue strategies", without (and before) digging into details depending on technical issues. The concept adopted in this phase are "topic", "relevant relation" and "group of topic"; with these three simple concept we can model the content elements of the applications, the relationships between these contents and the main access paths to these contents.
    - A "topic" is something that can be the subject of conversation between the user and the interactive application, i.e. it is what the application can speak about. If we take a museum web site as example, some topics may be the following: "Madonna and Child", "Raphael", "Presentation of the Museum".
    - We can make a distinction between "kind of topic" and "single topic": a "kind of topic" is the category of possible subjects of conversation (e.g. "painting", "artist"), a "single topic" is a topic with one instance in the application (e.g. "Presentation of the Museum", "How to reach us").
    - A "relevant relation" determines how the dialogue can switch from a "kind of topic" to another one; e.g. "authorship" is a possible change of subject relating any "author" to its "paintings".
    - A "group of topic" determines the entry points or access paths to the topics as possible subject of conversation; e.g. "Masterpieces" (a group of particularly relevant paintings).

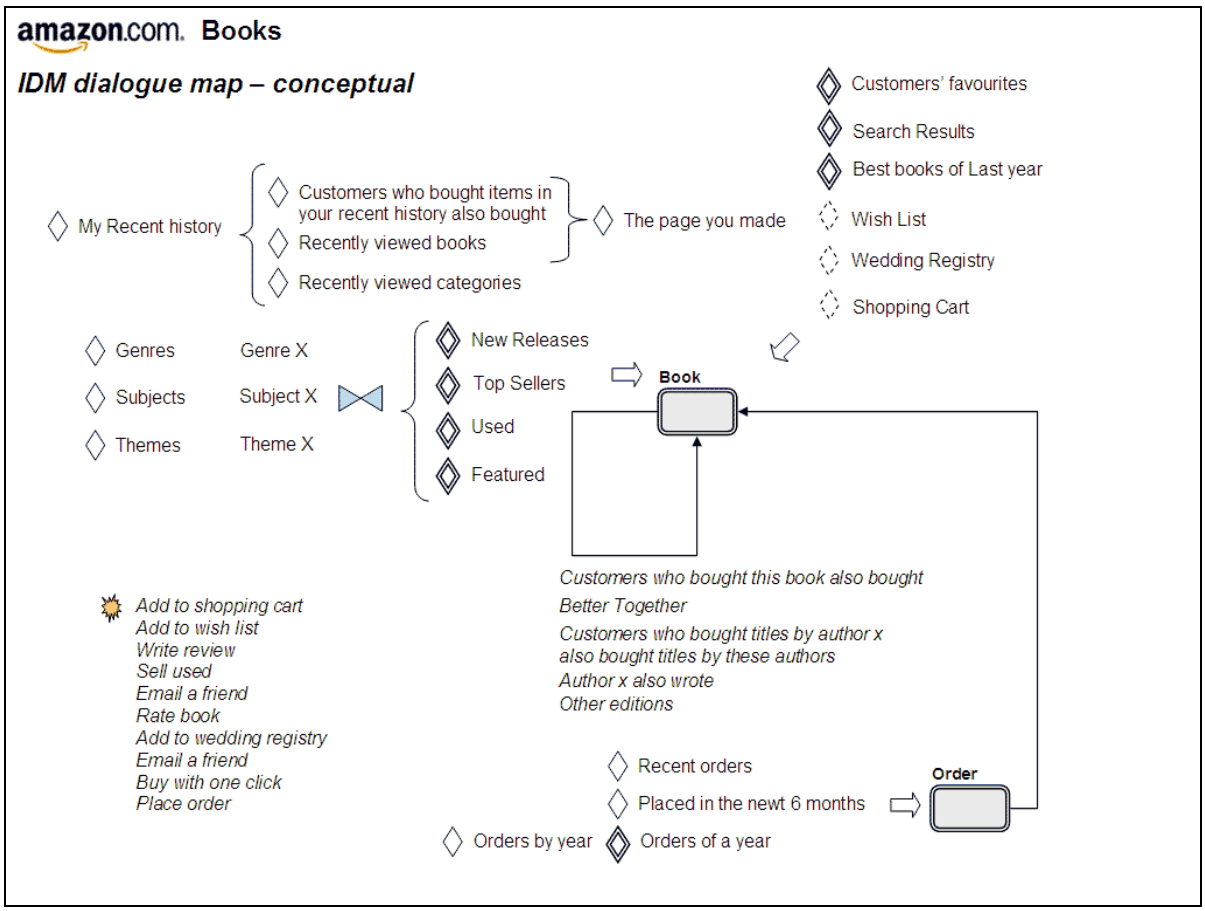

Fig. 3: IDM modelling of Amazon.com books web site

- A "parametric group of topic" determines a family of group of topics; e.g. "Paintings of a period X". Each "parametric group of topic" has a corresponding *higher-level* "group of topic" that allows to select the specific group of topic of interest; e.g. "All themes" allows selecting "Paintings of the XVI century".

– The Logical Design (L-IDM) can be seen as a detailed version of the conceptual design tailored for a specific channel (e.g. the Web). Here details are decided on the basis of a variety of channel-dependent factors: the type of device available on a given channel (e.g. screen size), the pointing device (e.g. keyboard, smart pen, mouse, scroller, audio input, touch pointers, eye-tracking pointers), the media which can be used (e.g. audio, visual text, images, graphics, or video) or the expected performance, and the typical scenarios of use (e.g. home or office desktop use, walking or standing contexts, mobile use on car, etc.). The concepts used in the L-IDM are the following: "dialogue act", "transition act", "introductory act" and "dialogue strategy".
  - A "dialogue act" is a unit of the dialogue within a topic; the content of a topic is either represented by a single dialogue act, or several of them. E.g. the kind of topic "painting" could be structured in two dialogue acts: a *general description* of the painting with a little image and the main data about the painting (the title, the author, the composition year, etc.) and a *big image* i.e. a big size version of painting image with just some basic information (title, author) and a short caption.
  - In some cases of changing subject no additional dialogue is need, since the dialogue can immediately switch, upon request. When the new subject is multiple an additional

part of dialogue is needed, that we call “transition act”. A “transition act” is a list of possible new topics (e.g. a list of paintings of the same author) that the user can choose to speak about.

- An “introductory act” is a piece of dialogue that allows the application (and the user) to consider the group of topic as a whole. It consists, in general, of an introduction followed by a list of the topics belonging to the group. “Introductory acts” are the unique starting points for the dialogue, in the sense that any dialogue starts with an introductory act.
- A “Parametric introductory act” is an introductory act corresponding to a “Parametric Group of Topic”. For a parametric group of topic we need a family of introductory acts (e.g. one for each painting period) and a further introductory act (e.g. the list of periods), holding the family together.
- To navigate inside the application and through the dialogue/transition/introductory acts one or more “dialogue strategy” can be defined. We can consider three different kind of strategies: “structural strategy”, “transition strategy” and “subject strategy”.
  - The “structural strategy” is the possible development of a dialogue for exploring a topic with more than one dialogue acts. What must be specified is the initial dialogue act, and the possibilities for changing the dialogue from one act to another one.
  - If we consider a transition act, its existence does not entirely solve all the problems: a dialogue sub-strategy (“transition strategy”) must be developed to explain the way a user can explore all the new topics (all the “paintings” made during the same period, in the examples above).
  - In the same way, only creating introductory acts does not solve the problem of “engaging a conversation” about the group of topics; there must be a “subject strategy” coordinating how the conversation can involve the introductory act and support the exploration of all the topics belonging to the group.
- In all the cases above, a set of navigation patterns can be used to define the dialogue strategies in a structured way.

– The Page design (P-IDM) defines the elements to be communicated to the user in a single dialogue act, i.e. in a single “physical” page of the application and crafts the actual pages containing the necessary elements to sustain the dialogue. For each page IDM defines:
  - the “main content” (i.e. the content of which dialogue act is presented in that specific page)
  - “structural links” to pages of dialogue acts of the same topic
  - “transition links” to pages of related topic (1:1) or to pages of transition acts (1:n)
  - “group of topic links” (next-previous in case of guided tour or to pages of introductory acts and the introductory act I came from)
  - “orientation info” (i.e. where I am in the application)
  - “landmarks” to relevant sections of the site (pages of single topics, or group of topics).

  Note that page design should not yet go into “wireframe” design (defining the visual page grid), neither into layout design (how elements are physically arranged in the grid), and neither into graphic design (actual rendering of the visual elements in the page). Whereas all these aspects contributes to define the visual communication strategy of the application, page design should provide the proper input to these activities by just specifying what are the important elements to be present in the page.

3.9 WAE

A UML Web Application Extension (WAE) has also been recently proposed by Conallen (Conallen et al, 2003) trying to cope with the modeling issues required for developing complex web applications. Instead of distinguishing between content, hypertext and presentation level, Conallen models web pages at the server side and at the client side by stereotyping UML classes. Stereotyped associations are used to represent hyperlinks and to model the mapping between client pages and server pages. Data entry forms which can be part of client pages together with their submit relationship to server pages are modeled by another class and association stereotype, respectively. Finally, there are also class stereotypes for Java Applets, Java Scripts, ActiveX controls and frames. Conallen does not discuss any behavior modeling apart from operations which can be defined together with the stereotyped classes and does not suggest any modeling phases. These technology-dependent and implementation-driven concepts are used to describe the so-called user experience design, which is then mapped into a proper logical architecture of the application.

3.10 Conclusions

Although, if properly used, current academic methods have the potentiality of enabling designers conceive high quality (say usable and effective) applications, they suffer, in our view, of some inefficiencies which contribute to a poor acceptance from the industrial environment. Modeling purpose is often only badly or vaguely specified with the respect of the overall development process. It is often claimed models are intended as support tool during the early analysis activities, but they are also intended to support the implementation activities. For example in both OO-H and WebML, models are used to automatically generate the running application. Cumbersome design documents are generally produced as output of the design activities. These documents risk to be hard to read and use both during the analysis activities (where a short and information intensive document is highly preferred) and the following implementation ones (where design choices need to be detailed). While separation of concerns provides a support during the analysis, in our experience with implementers, dividing design choices in several dimensions reduces the specification readability and thus its effectiveness for supporting the implementation activities. Proprietary concepts and notation are generally proposed by each method. Although the underlying design philosophy and principles are shared among most of the current proposals, every method for each design dimension introduces a variety of different concepts. These pay different attentions to the various peculiarities of Web applications. The availability of different, sometime complex, concepts, even if powerful on their own, increases the learning time and thus the negative perception of such methods for first-time users. Moreover, except a few cases, every method proposes its own notation reducing the knowledge interchange among them. Professional tools support is a crucial factor for enabling professional users exploit a design method. Suitable support tools can drastically reduce time needed for drawing the required models. Most of existing methods come together with ad-hoc, in-house made support tools. Except a few cases, these tools have not been thought for commercial use thus they result hardly acceptable from the industrial world. Companies are unlikely to adopt a method that lacks adequate tools support. Concerning the second category, that is, methods proposed in by the industrial world, UML is definitively considered the standard de-facto in the design practice. Referring to the web application domain, the only recognized method coming from the industrial environment is WAE, proposed by Conallen (Conallen, 2003).

4. STATE-OF-THE-ART ON USABILITY EVALUATION METHODOLOGIES

Within the field of usability methods it is possible to identify several approaches for evaluating web usability. Among them, the most commonly adopted are *user-based methods* (or *user-testing* methods) and *usability inspection methods* (or *expert reviews*) (Matera M. et al.: 2002).

4.1. User-Based methods (User testing methods)

User-based methods mainly consist of user testing, in which usability properties are assessed by observing how the system is actually used by some representatives of real users (Whiteside J. et al.:1988) (Dix A. et. al.: 1998). User-testing evaluation provides the trustiest evaluation, because it assesses usability through samples of real users. However, it has a number of drawbacks, such as the difficulty to properly select correct user samples and to adequately train them to manage also advanced functions of a web site (Matera M. et al.: 2002). Furthermore, it is difficult, in a limited amount of time, to reproduce actual situation of usage. This condition is called "Hawthorne effect" (Roethlisberger et al.: 1939): if the variable of the experiment are manipulated, it is possible that the productivity of the group observed decreases. Failures in creating real-life situations may lead to "artificial" conclusions rather then realistic results (Lim K.H et al.: 1996). Therefore, user-testing methods are considerable in terms of time, effort and cost. User testing is the main way for evaluating right away the look and feel of the interface, as it is possible to verify at "real-time" the reactions of the users.
Within the category of user-testing methods there are several techniques, the most important are:

- *Thinking aloud*
- *Contextual inquiry*
- *Focus group*
- *Interview*

4.1.1 Thinking aloud

During the thinking aloud test, the user should think aloud while performing some specific task with the system. By verbalizing his thoughts, the user allows the observers to know his opinions and feeling about the application. Verbal protocols are recorded concurrently or retrospectively. The subject is probed to verbalise problems that come up. After the recording of verbal protocols, the protocols are encoded according to a previously defined encoding scheme. Verbal reports can be interpreted if the processes by which they were generated are understood. Interpretation is based on the theory that human cognition is information processing (Newell & Simon 1972, Simon 1979). Cognitive processes and their structure account for the results of verbalisations. The accuracy of verbal reports depends on the procedures used to elicit them and the relation between the requested information and the actual sequence of heeded information.
Thinking aloud allows you to understand how the user approaches the interface and what considerations the user keeps in mind when using the interface. If the user expresses that the sequence of steps dictated by the product to accomplish their task goal is different from what they expected, perhaps the interface is convoluted.

Although the main benefit of the thinking aloud protocol is a better understanding of the user's mental model and interaction with the product, you can gain other benefits as well. For example, the terminology the user uses to express an idea or function should be incorporated into the product design or at least its documentation.

4.1.2. Contextual Inquiry

Contextual Inquiry is a specific type of interview for gaining data from the user. This technique aims at understanding the context in which the application is used. Contextual Inquiry (also known as "site visits") is basically a structured technique of observing and interviewing users. It is based on the core principle that understanding the context in which a product (or service) is used (or the work is being performed) is essential for user and customer oriented design. Using contextual inquiry, you visit the workplace of prospective users to see how they work. You observe all aspects that would help define a context for their work - and thus a context for the usage of your product or service.
Contextual Inquiry is adequate in situations where the subject domain is unclear or unfamiliar to the development team, and when the context of work may have a significant effect on the new product or service. For performing a Contextual Inquiry considerable investment of time and effort may be needed in order to elicit sufficient information from the users and the environment to be studied.
Contextual Inquiry follows many of the same process steps as field observations or interviews.

Contextual inquiry is best done by a group of researchers who develop a medium- to long-term relationship with a group of organisations who are interested in providing data. According to Holtzblatt et al. the relevant steps are the following:

- "Identifying the customer: identify the groups that will be using the new technology or are using similar technology, and arrange to access organisations within the groups that give a cross section of the (potential) market.
- Arranging the visit: write to the targeted organisations identifying the purpose of the visit, a rough time-table, and how much of the employees time will be taken up by the exercise. Ensure that some feedback from the day is possible before leaving. Ensure that the participating organisations understand how many visits you intend to make over the time period of the evaluations.
- Identifying the users: a software product will affect many people throughout the organisation, not just the management or the end users. Ensure that you understand the key users in the organisation whose work will be affected by a new system or changes in the current one.
- Setting the focus: select what aspects of the users' work you wish to make the focus of each visit, and write down your starting assumptions. Make a statement of purpose for each visit, and after the visit, evaluate to what extent you have achieved your purpose.
- Carrying out the interview / observation: stay with the selected users until you have managed to answer the questions you have raised in 'setting the focus'. Very often this may involve inviting the user to directly share and comment on your notes and assumptions.
- Analysing the data: the process of analysis is interpretative and constructive. Your conclusions and ideas from one round of observations are input to the next round, and an evaluation of the results so far should be one of the purposes of subsequent visits."

4.1.3. Focus Group

Focus group is a technique developed in the field of marketing research. The goal is to identify the problems of the application by means of discussions with groups of users. Focus group analysis is an informal technique that can be used to assess user needs and requirements. It can be applied at any time in the development process. In a focus group about 6-9 users are brought together over a period of about 2 hours to discuss whatever issues are of interest: new concepts, designs, prototypes, complete application. The moderator running the focus group is responsible for maintaining the focus of the group on the issues of interest following a pre-planned script. One of the main problems is that focus group meetings are demanding in terms of the number of representative users needed. It is preferable to run more than one focus group since the outcome of any single focus group session may not be representative. During the Focus group the moderator presents issues to be discussed in the focus group session. He tries to keep the discussion on track without inhibiting the free flow of ideas and comments from the participants. He ensures that all members of the focus group get to contribute to the discussion.

4.1.4. Interview

Interview is an informal technique for the investigation of the users' opinions about the application, e.g. subjective satisfaction, critical incidents, anxieties which are hard to measure objectively. It is a useful method for studying what features of the application users particularly like or dislike.

Three types of interviews can be distinguished: unstructured, semi-structured and structured interviews. The type, detail and validity of the collected information vary with the type of interview.

The validity of results varies with the experience of the interviewers. The interviewer needs domain knowledge in order to ask the right questions and there is always the risk of bias in what questions the interviewer asks and how the interviewee interprets them. Besides, Interviews are demanding in terms of the number of representative users needed. It is preferable to use questionnaires where possible. Because of the unstructured nature of an interview the result is just a report summing up the comments made by the subject in the interview.

4.1.5 Other User-based methods

Within the panorama of user-based methods there are other techniques, in particular:

- Co-discovery;
- Questionnaire
- USE Questionnaire
- Questionnaire for User Interface Satisfaction (QUIS)

Co-discovery

Co-discovery is an experimental method for investigating problems two users face while they are interacting with a system. The two users discuss the occurring problems or one user helps the other user to solve problems. Problems verbalised and discussed by subjects during a co-

discovery session are recorded. Afterwards the recordings of the verbal protocols are encoded according to a previously defined encoding scheme.

Questionnaire

The method aims at indexing the subjective appreciation of electronic information services or products. It looks for three components that together form an attitude: cognitive appreciation, emotional appreciation, and action tendencies the studied object might generate. It requires some form of dimensional analysis (Principle Component Analysis). Applying this method results often do not fit with the results of behavioural measurements. The expressed attitudes of persons are not particularly predictive of how they behave.

USE Questionnaire

The Use questionnaire measures subjective assessments of usability of a computer system. USE stands for Usefulness, Satisfaction, and Ease of Use, the three dimensions which emerged most strongly in the early development of the USE Questionnaire.

Questionnaire for User Interface Satisfaction (QUIS)

QUIS measures the user's perception of a human-computer interface with a questionnaire. Subjective ratings by users. Normally, the standard measures used within the QUIS are:

1. Overall reactions to the system
2. Screen
3. Terminology and System Information
4. Learning
5. System Capabilities

4. 2. Usability Inspection Methods

Usability Inspections methods is the generic name for a set of methods based on having expert evaluators inspect or examine usability-related aspects of a user interface (Nielsen J. et al.: 1994). The term *Usability Inspection* born within the fields of software engineering in reference to function and code inspections methods that have been used in software engineering for debugging and improving code (Ackermann A.F. et al.: 1989).

With respect to user-testing evaluation, usability inspection methods are more subjective, having heavy dependence upon the inspector skills (Matera M. et al.: 2002). The focus of usability inspection methods is on the usability related aspects of user-interface of interactive products and services. The objectives of this approach are bounded to the identification of some interface problems in an existing design, and then using these problems to make recommendations for fixing the problems and improving the usability of the design. This means that usability inspections are normally used at the stage in the usability engineering cycle when a user interface design has been generated and its usability (and utility) for users needs to be evaluated (Nielsen J. et al.: 1994).

The main advantage of inspection methods is the relationships between costs and benefits. In fact, performing usability inspection “save users” (Nielsen J. et al.: 1994), (Jeffries R. et al.: 1991) and does not require any special equipment and the inspector alone can detect a wide range of usability problems and possible faults of a complex system in a limited amount of time

(Matera M. et al.: 2002). For these reasons, inspection methods have achieved widespread use in the last years, especially in industrial environments (Madsen K.H., 1999). However, current usability inspection methods have a number of drawbacks:

- They focus on "surface-oriented" features of the graphical interface (mainly at page level) (Green T.R.G et al: 1996). Only few of them address the usability of the application structure, i.e., on the organization of both information elements and functionality;
- They are strictly dependent on the individual know-how, skill and judgment of inspectors, making a subjective process. Domain and application experience may improve the evaluators' performance.

The main inspection usability methods for hypermedia and web applications are:

- Heuristic evaluation
- Cognitive Walkthrough:
- Pluralistic Walkthrough:
- SUE (Systematic Usability Evaluation):
- MiLE+
- Content Evaluation

4.2.1 Heuristic evaluation

Heuristic evaluation is the most informal method; usability specialists have to judge whether each dialogue element conforms to established usability principles or not (Nielsen J. et al.: 1994). Heuristic Evaluation (created by Jakob Nielsen in 1994) is an inspection method in which one or several evaluators systematically inspect the user interface according to general usability principles (called "heuristics"), which describe the ideal characteristics of a usable interface. The evaluators examine the interface and verify its compliance with these heuristics.In 1990 and 1994, Jakob Nielsen, in collaboration with Rolf Molich, developed a very-well known list of 10 heuristics, which became general principles for user interface design and usability review. One of the main benefit of heuristics inspection – independently from the specific set of heuristics used – is that it provides a "guide" for the evaluators about where and what to look in an application and how to interpret its complexity. In this way, heuristics are a useful tool to "force" inspectors analyze the different aspects of the user interface, which are often overlooked without a supporting method at hand. However, some drawbacks should be also noted for heuristics-based inspection. Heuristics enable to carry out a "static" analysis of the application (i.e. to verify if it is compliant with given principles); however, this compliance does not guarantee that the application can effectively support user's goals and tasks. It may seem a paradox that an application with no content (empty pages) is fully compliant with the most known usability heuristics.

The principles given by Heuristic Evaluation are fairly broad and can be applied to practically any type of user interface.

The 10 Heuristic provided by Nielsen are:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom

4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognise, diagnose, and recover from errors
10. Help and documentation

Each measure describes the number of usability problems found for this usability heuristic (Nielsen 1994, p. 30).

1. "The system should always keep users informed about what is going on, through appropriate feedback within reasonable time."
2. "The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order."
3. "Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo."
4. "Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions."
5. "Even better than good error messages is a careful design which prevents a problem from occurring in the first place."
6. "Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate."
7. "Accelerators - unseen by the novice user- may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions."
8. "Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility."
9. "Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution."
10. "Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focussed on the user's task, list concrete steps to be carried out, and not be too large."

*4.2.2 Cognitive Walkthroug*h

Cognitive Walkthrough is an inspection method which focuses on the evaluation of the ease of learning of a user interface and learning by exploration. In a cognitive walkthrough an interface design is evaluated in the context of one or more specific user tasks (VNET5 Consortium: 2001). The evaluator(s) acts as if the interface was actually built and he (in the role of a typical user) was trying to accomplish the tasks. Each step the user (embodied by the inspector) would take is scrutinized: impasses where the interface blocks the "user" from completing the task indicate that the interface is missing something or has some usability problem.

4.2.3 Pluralistic Walkthrough

The term "walkthrough" is nearly synonymous with "storyboarding". Walkthroughs can provide usability data even if no user interface prototype or system is available. The walkthrough is conducted by identifying primary tasks for the application and stepping through those tasks, identifying usability problems along the way.

The purpose of bringing together various stakeholders is that each one brings a certain perspective, expertise, and set of goals for the project that enables a greater number of usability problems to be found

Three types of participants are involved in this inspection method: representative users, product developers, and usability experts. All participants in the walkthrough - representative users, product developers, usability experts - are confronted with hard-copy panels of screens in the order these would appear online in a real application. For each hard-copy all participants describe as detailed as possible the actions they would take in executing the task described in a scenario. Then the contributions from all participants will be discussed. The result will be usability problems detected during the walkthrough.

4.2.4 SUE (Systematic Usability Evaluation)

SUE is an inspection method for hypermedia applications. It provides a set of hypermedia specific usability criteria ("heuristics"), and a set of domain-independent tasks (called "Abstract Tasks") that inspectors should perform in order to systematically inspect the different aspects of a hypermedia application (Matera M. et al.: 2002). Abstract tasks can be executed on using a final application, a running prototype, or a set of design specifications. They are "inspection patterns", each one focusing on a specific feature of the application (e.g., the synchronisation of multimedia data, the navigation of a guided tour or a table of content, indexes, etc.). The inspector gathers the usability problems she identifies by performing each abstract tasks, in order to judge the usability of the overall application, or of some specific aspects. The use of abstract tasks makes the inspection activity more structured, better organised, and helps an organization to standardise and compare the inspection results of different inspectors.

4.2.5 MiLE+ (Milano-Lugano Evaluation)

MiLE+ (Milano-Lugano Evaluation method, developed in cooperation between Politecnico di Milano and University of Lugano) is the evolution of MiLE method (Triacca L. et al: 2003, 2004). MiLE+ is an experience-based usability evaluation framework for web applications that strikes a healthy balance between heuristic evaluation and task-driven techniques.

MiLE+ offers reusable tools and procedures to carry out both inspection and user testing within budget and time constraints. For these reasons, MiLE+ proposes two types of inspection activities namely ***Technical Inspection*** and ***User Experience Inspection***.

Before explaining the activities of MiLE+ it is important to underline that it employs general elements for performing its activities. These elements are:

- ***Scenarios:*** scenarios are “stories about use” (Cato, J.: 2001; Carroll J.: 2002), describing a typical user, one or more goals, and elements of the context of use (place, time, circumstances of use, etc.). MiLE+ uses scenarios as the driver for usability evaluation, because their role is crucial for an effective usability evaluation.
- ***Heuristics:*** as said in the Background and related works heuristics are usability guidelines/principles that allow the evaluation of an application. MiLE+ provides two sets of heuristics that should help the evaluation:

- ***Technical Heuristics:*** a set of heuristics enabling to evaluate the design quality (in all its aspects) and to spot implementation breakdowns. Technical Heuristics are organized in design dimensions (e.g. content, navigation, graphics) and associate each design dimension to a list of guidelines which help the inspector to analyze each dimension from a "design" perspective.
- ***User Experience Indicators (UEIs):*** There are aspects of usability which cannot be evaluated by those who are not final users. In other words, User Experience Indicators allow anticipating the potential problems that end-users may encounter during their experience with the website. Examples of UEIs are understandability, frustration, satisfaction, attractiveness, etc. Therefore, they allow the evaluation of each scenario's quality with respect to these user experience characteristics.

- **Usability Evaluation Kits (U-Kits):** to facilitate the inspection activity MiLE+ offers a set of reusable evaluation tools (U-KIT, the usability evaluation kit) and the guidelines for creating one. A U-KIT is a library of specific evaluation tools, which comprises a library of scenarios (User Profiles, Goals and Tasks) related to a specific domain, a library of Technical Heuristics and a library of User Experience Indicators.

Technical Inspection

The aim of MiLE+'s Technical Inspection is the identification of design problems and implementation breakdowns. The output of this evaluation is a number of "technical" problems that are application independent (e.g. the fact that the font size of a text is too small – graphic technical problem – it is a problem independent from the type of application). During this analysis the evaluator examines the web application taking into account a number of design dimensions, assuming the point of view of the designer and not of the end-user (like during the User Experience Inspection). The design dimensions are:

- *Navigation:* the website's structure
- *Content:* information provided by the application,
- *Technology/Performance:* technological performance of the application.
- *Interface Design:* this is a broad dimension that includes semiotics (this dimension will be widely discussed in paragraph 4), graphics (graphical design and layout) and cognitive (what the user learns about the application and its content)

During the Technical Inspection problems are discovered using the heuristics checklists (selected from the library of technical heuristics) and scenarios: these two elements compose the U-KIT for Technical Inspection. It is important to underline that the use of scenarios are not mandatory. Indeed, we do not evaluate the adequacy of scenarios, but they are useful for navigating with clear goals within the application (so the inspector can concentrate his evaluation on the most important parts of the website).

User Experience Inspection

The User Experience Inspection is a scenario-based evaluation. This means that the evaluator has to imagine stories of use. During this inspection the inspector has to examine the adequacy of the scenarios: in this sense the User Experience Inspection is application dependent. For this reason, he has to set-up the "User Experience" KIT tailor-made for the application under analysis. The KIT is composed by:

- ***Scenario library:*** for creating a domain's library the inspector has to interact with different stakeholders: the client, domain experts, end-users, etc. For example, in creating the library for evaluating a museum websites the inspector should interview the Director

of the Museum, he should organize a focus group with art's experts and a focus group with end users. Another complementary way for creating the library is called the "visioning technique" (Cato, 2001). The inspector has to imagine which ones are the main end-users, their goals and tasks: it is clear that this technique is more superficial but it can still generate reliable results.

- ***Library of User Experience Indicators***: during the User Experience Inspection the evaluator has to put himself in the "shoes of the (different) users". This means that he has to examine the relevant scenarios using some criteria called User Experience Indicators. These criteria are divided in three categories corresponding to the different types of user interaction experiences. These categories are:
    - *Content Experience Indicators:* measure the quality of user interaction with the content of the application.
    - *Navigation & Cognitive Experience Indicators:* allow the measure of how the navigation works and the cognitive aspects of the application meet the cognitive world of the user(s).
    - *Interaction Flow Experience Indicators:* permit the measurement of how the interaction with the application is appreciated by the users.

The User Experience Inspection is strictly related to the Scenario-based User Testing. Indeed, the main goal of the Scenario-based User Testing is to empirically validate or invalidate the results provided by the User Experience Inspection. During the test the user accomplishes several tasks belonging to the critical scenarios identified in the User Experience Inspection.

4.2.6 Content Evaluation

For information intensive interactive products, the approach to inspection can also adopt methods of content analysis and communicability evaluation. The objective of content analysis is twofold:

- inspecting the quality of content allows detecting quality breakdowns in the communication
- content evaluation methods suggest guidelines for designing usable content.

From a communication perspective, the standpoint of methods for content evaluation is focused on the belief that the "happiness" of a communication act must be assessed by a receiver's point of view. Therefore, especially when dealing with content (i.e. coping with the notion of meaning, sense and relevance), the inspector has to take into account that addressee as the starting point and the target of the whole communication effort. Content should not be primarily intended in its technical sense (e.g. image size, length of pages, colour of icons), but it should be addressed as a designed set of ideas and messages conveyed through structured interactive possibilities. The main methods in this fields are:

- **Content Analysis**: Content analysis offers a set of conceptual tools for assessing the effectiveness and the quality of communication of a web application (from navigation to content).
- **Content Evaluation:** Content evaluation of electronic sources relies on the same principles as evaluation of a print source. Content evaluation is performed with a checklist for the five criteria: authority, accuracy, objectivity, currency, and coverage.
- **Criteria for the Evaluation of Internet Information Resources:** The criteria for evaluating Internet information resources is an attempt to amalgamate and assimilate

criteria from several sources that can be applied for evaluating and selecting Internet information sources.

- **Internet Information Evaluation Form:** The criteria for evaluating Internet information resources is an attempt to amalgamate and assimilate criteria from several sources that can be applied for evaluating and selecting Internet information sources.
- **Quality of Internet Information Sources Criteria Questionnaire:** The criteria for evaluating Internet information resources is an attempt to amalgamate and assimilate criteria from several sources that can be applied for evaluating and selecting Internet information sources.

4.2.7 Other inspection methods

There are several other inspection methods. In particular:

- Formal Usability Inspection
- Inspection and Design Review
- Software Inspection
- Cognitive dimension framework
- User Action Framework

Formal Usability Inspection

The Formal Usability Inspection method was developed to help engineers to efficiently review the users' potential task performance with a product. The method is based on a formal inspection process consisting of six steps for the detection and description of usability defects. A formal usability inspection consists of one phase where the inspectors work alone. For each defined user profile and task scenario combination the inspectors take the role of the specific user and work through the tasks described in the task scenario. Usability defects are logged on defect logging forms. In addition a task performance model and heuristics are applied to detect defects. Afterwards, all inspectors come together to a logging meeting to aggregate their defects and to find more defects.

Inspection and Design Review

Inspection and Design Review is a general framework for user interface inspections which takes explicitly into account the purpose and the focus of the evaluation. The domain of concern and the depth of the inspection is determined before the inspection starts. Inspections are performed either individually or in groups. The inspection process can be more or less structured. The results are usability problems detected during the inspection and recommendations how to solve them.

Software Inspection

Software Inspection is a technique used to detect defects in software components or finished software products. The objective is to test the minimum requirement: Is the software (or a software component) free of bugs/errors? The domain of concern and the depth of the software inspection are determined before the inspection starts. The procedure for carry out the Software inspection is:

1. The Quality Manager checks if the software is ready for inspection and determines the objectives for the inspection.
2. The moderator plans and prepares the inspection on the basis of instructions received from the Quality Manager. He may also use information from previously executed inspections.
3. During a kick-off meeting the moderator explains the objectives of the inspection to the experts and provides them with the software to be inspected.
4. The experts test the software, log the defects they find, and prepare for the defect logging meeting.
5. During the defect logging meeting the defects found by experts are summarized. The severity of defects is assessed. Finally, a causal analysis of defects and solutions to prevent the most important defects will be performed.

4.3. Automatic methods

A third way for evaluating the usability of a web application is representing by automatic methods, which measure the usability by running a user interface specification through evaluation software. The literature (Nielsen J. et al.: 1994) suggests that this approach do not work, for the reason that, until this moment, it is very difficult to create a software that it is able to capture all the usability problems that refer different levels (cognitive, navigation, content…).
Most methods for evaluating web site quality assess static HTML according to a number of pre-determined guidelines, such as whether all graphics contain ALT attributes (Ivory, M., Hearst M.: 2002). Another example (Chi, E. H., Pirolli P. and Pitkow J.:2000) is represented by a simulation for generating navigation paths for a site based on content similarity among pages, server log data, and linking structure. Neither of these approaches account for the impact of various web page attributes, such as the amount of text or layout of links (Ivory, M., Hearst M.: 2002). In general usability aspects such as consistency and information organization are unaddressed by existing tools. In general, automatic methods are based on several sets of guidelines that are useful to measure page performances, to check the links' quality, for verifying the quality of HTML code, but some experiments (Ratner J., Grose E.M. and Forsythe: 1996) have shown that, for example, that HTML guidelines themselves have little consistency. However, automatic methods are a good complement to standard evaluation techniques (inspection methods and user testing) not a substitute.

4.4 CH specific heuristics and guidelines

Minerva is a network of Member States' Ministries to discuss and harmonise activities carried out in digitization of cultural content, for creating an agreed European common platform, recommendations and guidelines about digitization, metadata, long-term accessibility and preservation.
On March 2002 the *Minerva* project was launched with the support of the European Commission and the coordination of the Italian Ministry for Cultural Heritage and Activities. One of the working groups of Minerva was devoted to "Quality and accessibility of Cultural Institutions web sites". This Italian working group drew up a quality handbook for cultural websites developed by public institutions where quality criteria and methods for assessing cultural web applications are outlined [see: www.minervaeurope.org /publications/qualitycriteria.htm].

According to Minerva workgroup the goal of setting a quality framework is to break down quality into a series of criteria which are specific to cultural web sites. In Minerva handbook there is a great stress to definitions and to clarify concepts, areas and subjects ("cultural entity", "cultural web application", "goals of a cultural web application"). This effort is perhaps the most important outcome of the workgroup.

With regards to accessibility the handbook adheres to the guidelines drawn up by the WWW Accessibility Initiative. As regards usability Minerva workgroup proposes a list of criteria and recommendations in order to define the policy and strategies underlying the whole project of preliminary development of a CWA.
The handbook stresses three distinct but related aspects to the topic:

- The **definition of a policy of appurtenance** to new Web communities, thus permitting – given evaluation of the pre-requisites of quality – access to a specific domain name (cf. 1.3.1 and 2).
- The **adoption of strategies for coordination** of information flow within the CE and coordinated and organic use of the various channels of communication
- The **provision for planning procedures** which ensure efficient realization of Web Applications which adhere to the internationally recognized standards and regulations.

Quality criteria identified are:

- transparent
- effective
- maintained
- accessible
- user-centered
- responsive –
- multi-lingual
- interoperable
- managed
- preserved

As far as evaluation methods, Minerva handbook proposes only a checklist of checkpoints for web accessibility, but no methodologies.
The results concerning "Quality in Web Applications: general principles and operative proposal" cover the following design and evaluation issues:
Accessibility of contents

- Disability
- Current standards and the EU policies

Usability

- Definitions
- Principles

Criteria of Usability for Cultural Web Applications (CWA)

- Make contents perceivable
- Recognise that the site is a Cultural Web Application
- Recognise the aims of the site
- Gain a general impression of the site before proceeding to a detailed visit.
- Be able to exploit quality contents
- Presentation of Content
- Design a Functional layout

- Design Functional graphic elements
- Design Functional multimedia elements
- Site Navigation
- Searching

Patterns and the language of Patterns in Cultural Heritage

- Definitions of Design Patterns
- The Catalogue of Patterns
- How to consult Patterns
- An example of the use of the Catalogue of Patterns

5. STATE-OF-THE-ART ON ACCESSIBILITY METHODOLOGIES AND STANDARDS

Web Accessibility definition:
"To put the internet and its services at the disposal of all individuals, whatever their hardware or software requirements, their network infrastructure, their native language, their cultural background, there geographic location, or their physical or mental aptitudes."
Tim Berners-Lee, director of the W3C (www.w3c.org)

5.1 Web Accessibility methods, standards and legislations

The problem of how to make and validate accessibility features of web applications has recently gained much interest.

"Considers that the WAI initiative, which is voluntary in nature, should be strengthened to require, on a mandatory basis, all public websites of the EU institutions and the Member States to be fully accessible to disabled persons by 2003, which is the European Year of Disabled people; furthermore, calls on the EU institutions and the Member States to comply with the authoring tools accessibility guidelines (ATAG) 1.0 by 2003 as well, in order to ensure that disabled people can read webpages and also to enable them to manage the content of the webpages (content management);" (Excerpt from: European Parliament resolution on the Commission communication eEurope 2002: Accessibility of Public Web Sites and their Content)

W3C Consortium, that supplies the "strategic" guidelines for the web, has emanated a standard, based on documents prepared by associations of visually impaired people. The standard is composed by a set of guidelines helping web designers to better understand the main problems and solutions in developing an accessible website. Curently W3C accessibility guidelines are the only official technique for web accessibility design: new models and guidelines are going to be developed, but at the present there are very few tested and approved methodologies devoted to web accessibility. Furthermore, it is even more acknowledged that accessibility is strictly related to usability, in the sense that user-centered design and usability techniques often helps accessibility aspects and, on the contrary, an accessible website is often a usable website for anyone. Therefore, current web design and web usability techniques are going to be revised without defining new ad-hoc methods and models.
In the following sections, an introduction to the paradigm of interaction with a website through a screen-reader will be presented for better understanding the relevance but also the limits of the current standards. Then, it will be shown an innovative approach to web accessibility developed by the University of Lugano in collaboration with the Politecnico di Milano, having the aim to overcome some of the W3C limits. Finally, some consideration regarding the presented techniques and an outlook to future researches will be exposed.

5.2 How disable people access the internet

Whilst character-based interfaces offered blind people the extraordinary possibility to make use of their skills in using keyboards and interacting with software tools, graphic interfaces, implying

complex pages' layouts, many visual features and above all the use of the mouse have made their use of the many valuable resources offered by the Web a difficult and cumbersome task.
Developing separate Websites specially dedicated to this category of users is definitely not the right solution: first of all, not all the institutions would be willing to pay double costs to develop and also to keep updated two different Websites; a check of the multilingual versions of many Websites clearly demonstrates that usually the main Website is updated whilst its "foreign clones" are left behind, in terms of graphic, content, services, etc. Moreover, blind users themselves refuse being "ghettoized", rather claiming that a better design would enhance the efficiency and satisfaction of the Web experience for *any* kind of user (Theofanos & Redish, 2003).
Visually impaired people currently access the Web by using screenreaders. A screen reader is a software program that allows a blind person to read text on the screen and identify some graphics like buttons on a toolbar or icons on the Desktop. A person hears the information from a speech synthesizer or the computer's sound card. A screen reader also allows a person to control the computer using the keyboard rather than the mouse. Many of the keyboard commands that a blind person uses are the same keyboard commands a sighted person can use on their PC. A blind person also uses special keyboard commands that were created by the screen reader. Users interact with it by using keyboard commands, i.e. ENTER for selecting a link, TAB for going to the next paragraph, INS+F7 for having the list of all the links on the current page without listening to the rest of content, etc.
This means that there is a drastic shift in the interaction dynamics with respect to having users using visual supports while interacting with the website: content and navigation designed for being accessible through a visual channel ("seen on the screen" as it is on the web) should be re-designed or optimised for being accessible on an oral channel ("listened to").
The screen reader does have access to the html code of a page, not to the rendered version of it. This allows the designer of a web application to use the current standards of the web (e.g. Cascading Style Sheets) to organize the page in a nicer way for the listener keeping the graphical layout of a "normal" page. The standard does also provide a way to enrich the user experience of a website, allowing the designer to choose different voices, speeds and volume for different parts of the page. Unluckily this part of the standard has never been supported by modern browsers and therefore by the different screenreaders. While browsing the web with a screen reader the user lose information about the semantic of the page itself. A new family of controls has to be developed in order to convey those information to the user. A new generation of screen reader, supporting those meta-comments has to be developed too and because of the peculiarities of this approach that allows the user to listen to the webpage, rather than to the screen, from now on this new software will be called page reader. A page reader should convey to the user all the semantic information that are lost during the "translation" from graphic to speech: these information (like "this section is the most relevant", or "this is an ad") will allow the visually impaired user to actually enjoy the browsing experience. The most straightforward choice to store the meta-data into the page is to use specific tags. Just like in the CSS case the result will be a "semantic" stylesheet applied to the page.
Screenreaders' worth is clear; nonetheless, their limits. In particular:

- They read everything, including elements of HTML that are useful for visualization only (and do not convey relevant meaning to the listener).
- They have (by default, at least) a simplistic listening strategy, "top-to-bottom/left-to-right", making it difficult and boring to wait for the relevant piece of information. The

reader is invited to read aloud a page of a daily newspaper adopting the same strategy and measuring how long it takes until something relevant is read.

- They fail to convey the overall organization of the page, with the relative priorities of the different parts.
- They interleave the reading of content with the reading of links, with a total confusion for the listener. The listener can get the links' list (in alphabetical order), without the content, but s/he can't get the content without the links! In addition, even the list in alphabetical order is not effective; what if many links begin with the same word? Or if they're in an interrogative form, for example all beginning with "where can I find…"? Again, this means time and patience in waiting for the links' meaning to clarify, or wrong and time-consuming moves in the site (Theofanos & Redish, 2003).
- The selection mechanisms of the links are difficult and cumbersome. While in theory it is possible to "confirm" the selection while "listening" to a link, in practice, due to synchronization problems (of the audio with the current position on the page) it almost never works.
- Pages' layout and the "graphic's semantics" (that is, fonts' size and color, position on the page) are completely lost: the metallic voice of the screenreader will read one by one all the pieces of information of the page with the same emphasis and tone (the landmarks, the main content, the service links…), as if they all shared the same degree of importance.

Some of the problems of the screenreader are "technical", in the sense they can be (almost) mechanically checked, while some other problems are more "conceptual", involving design techniques and usability issues.

5.3 Current web accessibility techniques – W3C accessibility guidelines

The W3C consortium made public a first set of guidelines in May 1999. The second version of these guidelines (Draft 2.0) is currently under preparation (http://www.w3.org/TR/2004/WD-WCAG20-20041119/). It consists of 4 major guidelines prescribing that an application should be *perceivable*, *operable*, *understandable* and *robust*. For each of the four guidelines, *checkpoints* are defined. For each checkpoint (that are considered normative) *definitions*, *benefits* and *examples* (non normative) are provided. Checkpoints are classified either as "core" or "extended": to conform to WCAG 2.0, the Required Success Criteria of Core Checkpoints must be satisfied; the "extended" ones are additional checkpoints that may be reported in addition to Core conformance.

The W3C standards for accessibility has made the first fundamental steps to overcome the above problems and guarantee web access to visually impaired users. A set of guidelines have been defined and addressed to designers who want to make their site "accessible" for users with visual disabilities.

For example, a proper alternative text for each image is prescribed (the screen reader reads the alternative text so that a description of the image can be provided), and suggestions for correct contrast between the background and the texts are provided. Guidelines are also defined for designing tables on the web page that might be read by screen readers in a more meaningful way for the user.

Besides specific and detailed indications on in-the-small components of the page, guidelines for effective navigation and layout design are poor and often too vague. Especially with regards to

layout and navigation, many of W3C recommendations need to be interpreted and expanded in new, more detailed guidelines, affecting the content and the design of the site.
The guidelines are here presented, with some comments regarding the their acknowledged limits.

Guideline 1: PERCEIVABLE. Make Content Perceivable by Any User

1.1 [CORE] All non-text content that can be expressed in words has a text equivalent of the function or information that the non-text content was intended to convey.

This is a concern about content: the idea is that graphic and visual content should have a text equivalent. Still, what equivalence means is very difficult to define (see *figure 1*): which words are equivalent to a painting, an image or a map? Should the text convey the look, the semantics, the emotion, or what else? It is obvious that mechanically satisfying the guideline will not ensure "real" accessibility.

Figure 1: from the Museum of Modern Art Website www.moma.org - is it a text equivalent to the picture's meaning?

1.2 [CORE] Synchronized media equivalents are provided for time-dependent presentations.
Time dependent presentations, with audio synchronized to changing images, for example, are clearly a major problem for blind users.

1.3 [CORE] Both [information/substance] and structure are separable from presentation.
This is an important guideline, the potential meaning of which is much deeper than the W3C guidelines seem to imply. We should remind the reader that the key problem lies with HTML where presentation is intermingled with content. In addition, the guidelines focus on presentation details (which are important) and substantially neglect the problem of presentation strategy (which is even more important than details). Furthermore they overlook the fact that for "reading aloud" a page a presentation strategy is necessary: an "oral strategy" very different from the one based on visualization (as it is the one commonly used for Web pages).

1.4 [CORE] All characters and words in the content can be unambiguously decoded.
This a technical requirement, necessary and, in a sense, obvious.

1.5 [EXTENDED] Structure has been made perceivable to more people through presentation(s), positioning, and labels.

This is a very ambiguous, and in a sense, incorrect guideline. It is (practically) impossible and (above all) useless to attempt to describe with words the "look" of a Web page. The reader may try this simple experiment: try to read the page of a daily newspaper to someone else. Very likely the reader will try to read aloud the semantics (e.g. "the most important news is… the second news is…") rather than trying to describe the visual aspects of the page. So the key point is to take a different point of view: a Web page holds a deep semantics, that is translated into a visual presentation. In order to make a page readable the best option is to start again from the semantics, not from the visual presentation.

1.6 [EXTENDED] Foreground content is easily differentiable from background for both auditory and visual default presentations.
In this checkpoint we spot again what we think is a major problem of the W3C guidelines: they focus on the symptoms neglecting the causes. The visual communication provided by a Web page is a mixture of background (same for each page) and foreground (different for each page): the overall semantics of the page, conveyed by background and foreground, must be translated into an "oral" communication.

Guideline 2: OPERABLE. Ensure that Interface Elements in the Content are Operable by Any User
2.1 [CORE] All functionality is operable at a minimum through a keyboard or a keyboard interface.
This is a necessary and obvious requirement, very important for users with operational disabilities.

2.2 [CORE] Users can control any time limits on their reading, interaction, or responses unless control is not possible due to nature of real time events or competition.
This is an important and necessary requirement. Our observation is that the corresponding implementation can be very difficult!

2.3 [CORE] User can avoid experiencing screen flicker.
We do not question the checkpoint, but it seems to be rather specific and too detailed: it could have been combined with other ones.

2.4 [EXTENDED] Structure and/or alternate navigation mechanisms have been added to facilitate orientation and movement in content.
This is a requirement concerning interactive content: every interaction provided by visualization and pointing mechanisms (e.g. the mouse) should be also made possible with different mechanisms. Important requirement, but difficult to implement; also we should work (in the research community) not at the mechanical reproduction of a visual interaction for a blind user, but to an "equivalent" solution. In other words, if normal sighted users get some "message" from a visual interaction, we should try to deliver (with different means) the "same message" to blind users, rather than trying to reproduce the interaction.

2.5 [EXTENDED] Methods are provided to minimize error and provide graceful recovery.
This is an obvious, but quite vague guideline. It is a feature desirable for all kind of users, although users with disabilities need to be especially "protected".

Guideline 3: UNDERSTANDABLE. Make content and controls understandable to as many users as possible

3.1 [CORE] Language of content can be programmatically determined.

Changes of languages are more easily understood with visualization (also for visual clues as, for example, use of different fonts) than by listening. We have experimented how difficult it is to listen to a sudden change in the language being used. Beside technical details, we think that change of languages should be banned, unless if forced by a quotation.

3.2 [EXTENDED] The definition of abbreviations and acronyms can be unambiguously determined.

Again we have realized that while looking to acronyms is "usable", listening to them makes very hard life for a user, if he can't look at the page. We think that acronyms should always have an alternative text, just like for images.

3.3 [EXTENDED] Content is written to be no more complex than is necessary and/or supplement with simpler forms of the content

This is a simplistic guideline. The problem of tuning content to the "profile" of the user is a standard one, and it has nothing to do with disabilities: a good application should always provide content of the proper level for all the different members of the intended audience.

3.4 [EXTENDED] Layout and behavior of content is consistent or predictable, but not identical

Again this is a true, but simplistic, checkpoint. Moreover, for visually impaired users, the visual layout has nothing to do with the "audio" layout: therefore the suggestion of putting navigational elements always in consistent locations (required success criteria for checkpoint 3.4) is useless. It would certainly be more important to tell the designer how to shape content and navigation patterns in a consistent manner.

Guideline 4: ROBUST. Use Web technologies that maximize the ability of the content to work with current and future accessibility technologies and user agents

4.1 [CORE] Technologies are used according to specification.

The use of "unofficial" features of technologies must always be avoided, not just for users with special needs.

4.2 [EXTENDED] Technologies that are relied upon by the content are declared and widely available.

Availability of the technologies required for using the application is again desirable for all kinds of users, not just for the ones with special needs.

4.3 [EXTENDED] Technologies used for presentation and user interface support accessibility or alternate versions of the content are provided that do support accessibility.

This is a dangerous guideline: if the goal is understandable, we should also realize that current technologies for accessibility (e.g. current screenreaders for blind users) are not fully satisfactory. Technologies for accessibility still need a great impulse, and further research needs to be pursued. Freezing the solution to the technologies available today is very dangerous.

Let us finally summarize our comments about the W3C guidelines:

- *Guideline 1: PERCEIVABLE. Make Content Perceivable by Any User*

Some detailed guidelines are absolutely correct. But there is something confusing (if not wrong) about the presentation: apparently the guidelines fail to understand that the semantics of the page should be the starting point, not the way the page itself is being visualized.

- *Guideline 2: OPERABLE. Ensure that Interface Elements in the Content are Operable by Any User*
  We do agree with most of the recommendations, which in general are more important for users with operational disabilities, with respect to users with visual disabilities.
- *Guideline 3: UNDERSTANDABLE. Make content and controls understandable to as many users as possible*
  This is the weakest part of the guidelines, vague and not usable, with the exception of the references to languages and acronyms, which are clear. There is a total lack of references to design principles and to semantics that should be the most important factor in guidelines concerning understandability.
- *Guideline 4: ROBUST. Use Web technologies that maximize the ability of the content to work with current and future accessibility technologies and user agents*
  These guidelines are concerned with issues so general, that the specific concern for users with special needs is unclear.

5.4 Automated techniques for detecting web accessibility problems

It is becoming more common for Web site Authors/managers to implement the use of accessibility verification, remediation and repair tools to assure site accessibility. These solutions can assist in decreasing IT overhead and human resources for development projects. This section will deal with the types of tools available and how they should be applied to your testing process.

Verification tools
An accessibility verification tool is a software solution or a hosted service solution that allows you to test a page that you are working on or a group of pages of a (Logical or Physical) Web site for compliance with the accessibility standards. This technology can be instrumental in developing accessible content quickly and at a greatly decreased cost. The solutions are available in many different forms:

- **Desktop Software** Runs on a client desktop and runs with or without user interaction. Desktop software is generally suited for testing of smaller workgroup sites Web based and local pages. It is generally not used to disseminate information to a team.
- **Server Software** Server based solutions generally run without requiring any user intervention except for initial configuration. Additionally, once configured or for initial configuration there is general a “Web-Based” interface so that after installation there is no requirement for access to the console of the server.
- **Hosted Services** A Hosted service is similar to the Server software; the main difference is that there is no hardware requirement since the server software is installed and maintained and the service providers’ facilities.

Remediation (Repair) tools
Remediation tools are always controversial to HTML developers. The reason is the author fears that the tools will invalidate their HTML by performing bad modifications. However, repair tools can be valuable in the management of Web site accessibility.

- The remediation solution you choose should be able to work with files that are developed or stored on Windows, Unix (Linux), or Apple based systems.
- The remediation system should include full support for spell checking any alternative text representations entered to comply with accessibility.
- The remediation system should allow Web teams to collaborate on repairs for single or multiple pages.
- The remediation tool should learn the meanings of input elements, objects, and any element requiring alternative text or element content.
- The remediation tool should allow values to be edited in a library environment outside of the repair process.
- The remediation tool should allow for automated repairs where possible.

If a repair (remediation) tool is implemented properly it can prove to be essential in the process of making a site accessible. Additionally, a good repair tool decreases the training requirement for your Web team as well as decreasing the likelihood of errors.

Monitoring tools
Remember that because Web site content is dynamic - it is by nature always changing. That means that an important piece of any accessibility strategy is a solution that will let you ensure that your sites remain in compliance on an ongoing basis. From a site wide perspective companies should implement unattended services either hosted or placed on their internal servers that will, once configured, constantly monitor Web sites or portions of the Web sites and alert responsible parties if there is a problem that brings them out of compliance.

What automated tools cannot do
An automated testing tool can be used to test your site or groups of documents in an unattended manner once they are configured. It is very important to remember that “NO” tool alone can validate the absolute accessibility of your web site. However a good tool can identify a majority of what needs to be verified visually. Additionally a good tool will let you know what pages do not need to be verified visually, based on the absence of elements that require visual verification. Remember: You will still need to assure that all visual checkpoints identified by the solution are accessible.

5.5 Research studies towards an usable accessibility - The WED technique

The WED accessibility technique has been developed by experts and scholars from different areas: linguists, usability experts, communication scientists, web designers and engineers. Usability experts record (by means of a video camera and the thinking aloud method) sessions of use of "information intensive" Websites (such as Museum Websites); linguists and communication experts interpret them in the light of existing dialogue models and linguistic principles, highlighting their special characteristics, and the analogies and differences with respect to comparable natural dialogues. Web designers use the "understanding" of both types of dialogues in order to adjust design methodologies and in order to build interactive applications, based on the "oral channel", rather than on visual support.
The WED "dialogic" approach stems from an observation and a basic assumption:

- *Observation*: visually impaired users can't look at a screen; therefore the interaction must switch from the visual to the oral channel.
- *Assumption*: the interaction between a human being and a Website can be interpreted in terms of a dialogue (although a very peculiar one!)

In WED technique a web session consists of the user getting pages, as a consequence of his/her "clicks" on the same pages. A Web session is interpreted as a sort of dialogue between a human being and a machine: the machine's conversational turns consist in offers of content/ interaction; the user's turns consist in the selection of an offer (by clicking or performing some equivalent action).
According to communication theories, WED considers the designer of a Web application has a very powerful role: s/he sets the boundaries for communication and creates a stock of signs that the users may activate (potential "dialogues").
These considerations are the ground for modeling and designing the interaction between a Web application and its user as a particular kind of dialogue. The designer of a Website tries to imagine all the possible interesting conversations for the user and provide navigation mechanisms in order to make them possible. The designer thus plays a crucial role in the dialogue process because the range of possible interactions available to the user is actually defined by his intentions, expressed through the content, the navigation and interaction capabilities offered by the Web application.
Since a user experience is a dialogue, a Website can be considered a form of "dialogue generator", i.e. a device capable of supporting several different conversations with different types of users.
As previously stated, even if a Website complies with the W3C guidelines, it can offer a very ineffective dialogue, not usable in practice by a blind user. WED wants to find design strategies and solutions (in terms of content structure and navigation capabilities) able to consider and solve the needs of visually impaired users.
In the above overall scenario come the questions: "How should we design a Web application in order to generate successful dialogues with its users? How should we consider the needs and limits of a particular category of users, such as visually impaired people? The WED research effort tries to answer these questions by observing traditional human-human dialogues, in order to grasp recurrent dialogic strategies used in a traditional oral interaction and interpret them in Web design terms. Indeed, there are some synergies between dialogic and Web design theories: in dialogic theories we have concepts and models that help planning what to say (the so called

inventio in ancient rhetoric, that is, the collection of all the ideas and pieces of information), which structure should be given to the content (the so called *dispositio*, the ordering of the elements according to the overall text's strategy), how we want to tell it (*elocutio*, the adequate wording of the meanings to be conveyed by the message) and how present it (*actio*, the actual performance) with respect to the audience considered (Cantoni & Paolini, 2001; Di Blas & Paolini 2001). Concepts used in Web design techniques such as W2000 (UWA Consortium, 2001) are very similar in the purpose: there are concepts for describing what we want to say (the so called *hyperbase design*), how to reach the information (the so called *access structure design*), how we want to tell it and which order we want to give to the different elements (the *navigation design*) and how we want to present it to the audience (the *publishing design*).
Web design techniques make a sort of *separation of concern*, for better understanding a Web application from different viewpoints and levels. These methodologies help designers in planning and effectively shaping a Website in all its complex communication elements, with a clear view of all the interactive mechanisms that stay behind it.
The WED Project found in W2000 methodology a very interesting ground for research, since many of the concepts used are easily applicable and comparable with dialogic theories. Let us consider, for example, the navigational dimension of a Website. In W2000 there are three possible navigational contexts: *structural navigation*, in which the user explores the pages belonging to the same "topic" (e.g. the pages corresponding to the same "painting"); the *semantic navigation*, in which the user navigates from one topic to a semantically related one (i.e. from a page describing a painting to the pages describing the "author"); *collection navigation*, in which the user explores a group of topics (e.g. all the paintings of a certain period).
Each navigational context can be described in a dialogic perspective. In *figure 2*, for example, an example of modeling of a page of the Oscar Awards Website (www.oscar.com) is shown, using W2000 (UWA 2001) notation. The design of the pages is straightforward: there is a page where a list of the winners is presented and the user navigates from the list to any of the winner actors.

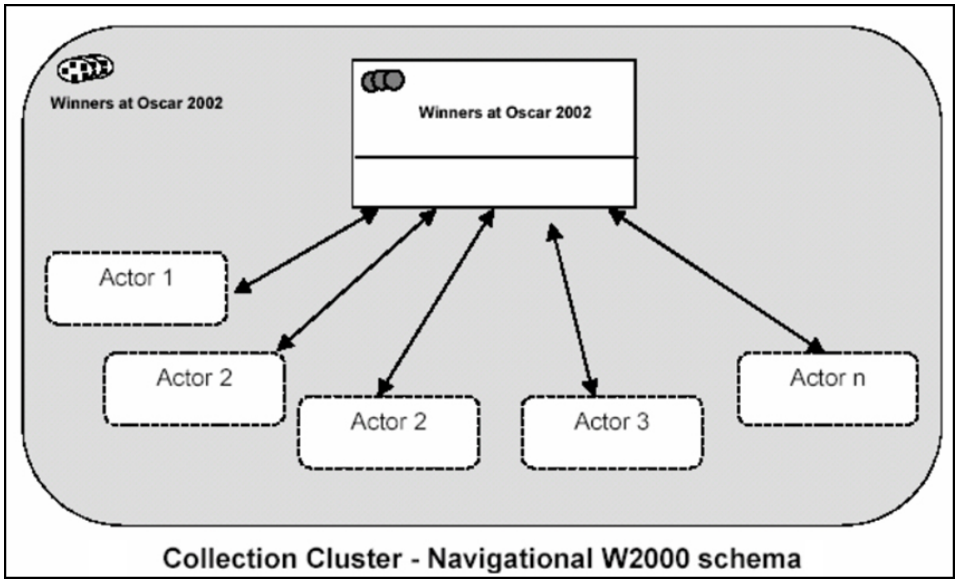

Figure 2 – Modeling of the 2002 Oscar winners Web page – www.oscar.com

From a dialogic perspective, here we are describing the introduction to a set of possible topics of conversation. In a traditional conversation, it would correspond to the question: " Which winner do you want to talk about?"

By considering this simple example, a lot of deeper design questions arise: “when the user selects a particular winner in the list and gets access to the details, how can she/he select a new one?” “Should she/he go back to the list, or should the Web application “guide” her/him and suggest what might come next?” In dialogic terms, if we consider partner A the application and partner B the user, the first solution could correspond to give the partner B the possibility to make a question to partner A like: “Could you repeat the names of the other winners you mentioned before?”. The second solution could correspond to the offer of partner A: “If you want, I can tell you about another winner: winner x”. Both from a dialogic and a hypermedia perspective, these strategies are very different and the result is a different interaction between the two partners.
Other crucial question is: what is the best way to organize the list, in order to make it effective? What criteria should be followed? Should items be listed in alphabetical order? Or should the names of the movies be used? How many items should be stored in the list? Should the list be split into sub-lists?
A designer should provide different answers according to the type of “channel” being considered: a purely oral dialogue requires different and specially tailored solutions with respect to a “visual” dialogue. If we can rely upon visual aids, we can display a list of 20 items, whilst if we are using the oral channel only that very same list becomes unusable: we would never list 20 names of awards winners in a normal conversation, asking our partner “which one do you want to talk about?”

WED technique is based on the assumption that oral dialogues use different strategies than visually supported ones; WED technique is the first step going beyond the consideration of accessibility design as a set of technical issues. If dialogue strategies are involved – in particular when an oral interaction is preferred to a visual one - than accessibility should be related to design and usability issues, in order to find viable solutions.

5.6 Conclusions and expected future techniques

Current standards and techniques provided by W3C do not address at all the question whether the application is actually usable (at least in a decent manner, and possibly in an optimal manner) by users with disabilities. As a matter of fact, it is known that it is possible to build applications fully compliant with W3C guidelines, and still unusable by users with disabilities. This situation is "politically" very dangerous since in several European countries (and Italy is a clear example of this) pieces of legislation are being approved, which make accessibility mandatory. If this is a positive aspect on its own, it becomes very negative when we consider that many bureaucracies (and the Italian one is an example) will take the easy way: "accessibility = W3C guidelines". What is likely to follow is easy to guess: "certifiers" (automatic or not) will certify that an application is compliant with the guidelines; the bureaucracy will be satisfied, certifying that the application is compliant with the law; the application developer will be satisfied since accessibility has been "achieved" at a reasonable cost; disabled users will be denied effective access to the Web application as before, since it won't be actually usable for them.
Web accessibility is strictly related to web usability: if we say that an application must be usable by all users, then users with disabilities must be included too. If this is deemed to be too difficult, then designers and developers of Web applications must clearly and carefully define which "user profiles" we mean to take into consideration and/or which ones not.

PROPOSITION 1: each Web application should clearly state which categories of users with disabilities have been taken into consideration (or not) as a target for accessibility.

The guidelines being proposed (version 1.0) or being considered (version 2.0) by W3C, within the WAI initiative, deal with a specific class of problems affecting usability: problems stemming from a bad use of technology. They do not address at all the question whether the application is actually usable (at least in a decent manner, and possibly in an optimal manner) by users with disabilities. As a matter of fact, it is known that it is possible to build applications fully compliant with WAI guidelines, and still unusable by users with disabilities. This situation is "politically" very dangerous since in several European countries (and Italy is a clear example of this) pieces of legislation are being approved, which make accessibility mandatory. If this is a positive aspect on its own, it becomes very negative when we consider that many bureaucracies (and the Italian one is an example) will take the easy way: "accessibility = WAI guidelines". What is likely to follow is easy to guess: "certifiers" (automatic or not) will certify that an application is compliant with the guidelines; the bureaucracy will be satisfied, certifying that the application is compliant with the law; the application developer will be satisfied since accessibility has been "achieved" at a reasonable cost; disabled users will be denied effective access to the Web application as before, since it won't be actually usable for them.
We think that researchers and practitioners working on this sensitive area of accessibility must fight in order to avoid this development.

PROPOSITION 2: a "manifesto" (signed by researchers and practitioners) should be used to make clear to all the politicians and bureaucrats, world wide, that satisfying the WAI guidelines does not mean at all that an application is accessible. Therefore it also should be made clear that "accessibility certification" can't be dealt with in a superficial manner.

If we equate accessibility to a usability problem, we acknowledge that ensuring it and checking it is more difficult, less automatic, and more debatable with respect to the expectations of those who rely on technical guidelines as the "solution". We can contribute to accessibility in two possible ways (with analogy to what has been done for usability): improving the way we check accessibility and developing tips and guidelines (based on best practices) in order to help developers in achieving it (at some level difficult to assess formally).

PROPOSITION 3: researchers and practitioners should work out a common way to carry on tests for assessing whether an application is accessible, to what degree and for whom; these "usable accessibility" tests must be based on a shared set of check lists and "assume" compliance with WAI guidelines (that can be checked separately).

PROPOSITION 4: researchers and practitioners should work out a set of "best practices" for designing and implementing really accessible applications (not nominally accessible); these best practices are a necessary complement to WAI guidelines.

If the above propositions are quite general, we can now draw the conclusions on our specific research, which is confined to a specific kind of disability, blindness, and to a specific technology, screenreaders and alike.
Within these limits we describe our current achievements and the broad lines of our future research:

- *Scientific background:* we believe that it is wrong to start from a Web page, conceived for being looked at, trying to make it readable. It is better to start from a step before: the "semantic of what is being said". This content must be delivered through an "oral channel", as opposed to the page, which is based upon a "visual" or "multimedia" channel.
  In addition we have found out (from empirical evidence and from linguistic literature) that the overall "dialogue strategy" is different, if the oral channel is being used, with respect to a channel with visual support.
  We are working, as far as basic research is concerned, in two promising directions: an empirical work of comparison between human-human oral dialogues, and human-Web (visually supported) dialogues. The development of a "dialogue model" (based upon semantics, rather than on syntax or rhetorical schemas), capable of capturing the essence of both types of dialogues, is what we are working at, together with a group of linguistic researchers.
- *General design*: we have already revised our previous design methodology (W2000), coming up with IDM - Interactive Dialogue Model. IDM is a tool (set of concepts and notation) to design an interactive application in a "conceptual", manner, i.e. independently from the specific channel that will be used for delivery. In a second stage the application will be "transformed" according to the need of the specific channel (oral, visual, ...).
  Although when the Munch's Website was developed, IDM was not fully defined, its basic principles were already there and were actually used; the result has been a very usable design, i.e. a structure of the application where consistency and self-evidence were emphasized. The benefit for the user is that s/he can easily understand the structure of the application, and how to move around.
  We are currently working on this notion of "usable design", that in our opinion lays at the very heart of true accessibility: if the user can understand the design and the motivations behind it, s/he will find the application more "natural".

- *Presentation strategy*: an oral presentation is radically different from a visual-supported presentation. We therefore came to the conclusion that it is useless to start from the page (from its "look") and to try making it accessible.
  Our goal is therefore to develop guidelines for an effective "reading strategy", based upon the intended semantics and the "raw content" of a page, rather than upon its look.
  A reading strategy can be considered at different levels of granularity: a section, a page, a group of pages, etc. We have found some simple rules, that we have already applied for Munch's Website, but much more research is needed.
- *Anaphoric strategy* (i.e. how to "go back"): one substantial contribution of linguistics to our research has been the recognition that a number of problems for accessibility stems from the practice, for the Web, of forcing the user to "go back" to already visited pages. This practice is ineffective for "normal users" and devastating for blind users (who must go through the whole page before getting to the point of interest). We have also understood the analogies between "going back" and the practice of "syntactic anaphora", as defined in linguistics (Di Blas 2003b).
  Our research works in three directions: trying to deploy navigation strategies that minimize the practice of "going back"; trying to improve the mechanisms implementing syntactic anaphora; trying also to implement mechanisms of semantic anaphora.
  The first two directions were somehow already considered for the Munch's Website, but we need to improve, in a number of ways, the solutions devised there. Considering semantic anaphora, i.e. moving back "to content" rather than to pages, is new and we need to break some new ground.
- *Screenreaders*: screenreaders have a basic limitation: i.e. they are not conceived for implementing an explicit reading strategy. For the Munch's Website we had to recur to a number of "tricks" in order to force the screenreader to implement what we had in mind.
  The strategic solution, that we are aiming at, is different: the reading strategy should be explicitly defined (at least at page level); the reading strategy should be "represented" in some ways (e.g. through "reading tags" or "reading instructions") in the page; a new generation of tools, "page-readers" should be used to implement the reading strategy.

The overall conclusion is that accessibility is scientifically challenging but also a socially relevant issue involving disadvantaged users. New tested and acknowledged techniques should be found, ensuring the development of accessible but also usable websites. Furthermore, new standards for accessibility evaluation should be proposed, testing not only accessibility in a technical manner but also the user satisfaction.

6. QUESTIONNAIRES

In this section we report the design of a questionnaire-based study we are carrying on within EPOCH, addressing two main goals: On the one hand, to investigate the current usage of design and usability evaluation methodologies, by identifying the most common methodologies and practices currently adopted by cultural institutions conceiving and evaluating their digital applications; On the other hand, to investigate the design and usability methodology needs, i.e., the actual desiderata by design or evaluation methodologies users (e.g., application analysts, usability experts, application developers, designers). The ultimate goal of the latter activity is the identification of the acceptability requirements of a design and of a usability evaluation methodology for cultural applications, in order to understand the characteristics that should be provided by a design or a evaluation methodology in order to be accepted and effectively used in the current practice of cultural multimedia projects. This work paves the ground for the development of guidelines, heuristics and best practices concerning design and evaluation of cultural multimedia, and for the identification for the criteria that a design methodology and that a usability methodology should satisfy in order to be recommended and accepted as a standard at the level of the EPOCH network and in at a broader level.

The questionnaire that we are using in our research is composed by two main sections:

- *investigation of the existing usage* of design, usability, and accessibility methodologies for cultural applications: which methodologies are actually used, what are their peculiar characteristics, in which phase of the application life-cycle are this methodologies used, etc.
- *investigation of the methodological needs* of CH application "stakeholders" directly or indirectly exposed to design, usability, or accessibility issues: our approach is to hypothesize a set of potentially important requirements for a usability model, asking respondents to judge their relevance.

The persons who are requested to fill the questionnaire are identified in the following stakeholder profiles: project managers of cultural heritage projects, usability experts, domain experts, application designers, requirements analysts, application developers with a significant experience in the development of cultural heritage applications, content mangers and producers in the field of cultural heritage and maintainers of cultural applications.

The questionnaires have been internally evaluated by the partners involved in this study (POLIMI, UNISI, IBC), and revised several times to improve its goals, clearness, and usability. It has been also tested with over 10 students of the Master in Technology Enhanced Communication for Cultural Heritage (held in Lugano in the first semester of the academic year 2004-05) and over 30 students of the graduate class in Multimedia Systems at the Faculty of Industrial Design of Politecnico di Milano (Track: Cultural Heritage Communication).
The questionnaire is currently available on-line on the EPOCH web site, and all partners have been requested to fill it. At the same time, the questionnaire will be mailed to a wide set of cultural institutions, by exploiting the existing cooperation with other EC funded projects in cultural heritage (e.g. MINERVA+), the contacts already established by University of Lugano in the context of the Master in Technology Enhanced Communication for Cultural Heritage, or the participation to international conferences (e.g., ICHIM, VAST, Museums and the Web, ….).

6.1 Questionnaire "EPOCH Survey on design methods for Cultural Heritage applications"

Questionnaire introduction

This survey is carried on in the context of the EPOCH project (http://www.epoch-net.org/) - workpackage 4.2.

EPOCH, funded by the European Commission under the Community's Sixth Framework Programme. (contract no. IST-2002-507382), is a **network of about a hundred European cultural institutions** joining their efforts to improve the quality and effectiveness of the use of Information and Communication Technology for Cultural Heritage (CH).

This questionnaire aims at to investigating the **current usage of design methodologies**, identifying the most common approaches and practices currently adopted by **cultural institutions** in the design of digital applications. We also attempt to identify some critical **requirements for CH design methods**, by understanding the needs and expectations of the various professionals involved in the design of multimedia applications in the specific field of cultural heritage (e.g., application analysts, domain experts, project managers, information architects, interface designers, application developers). In particular, we want to explore the characteristics that should be provided by a design method in order to be *accepted* and *effectively used* in CH projects.

If you are involved in any aspect of the development of a CH digital application (project management, requirement analysis, design, content management, prototyping, implementation, evaluation), please help us in this research!

The information you will provide – together with the data collected from other questionnaires - will work pave the ground for defining **development of guidelines and best practices for the design** of cultural multimedia, and for the identification for the criteria that a design methodology should satisfy in order to be **accepted** and effectively **adopted** in the cultural heritage world.

The data collected by our study will be treated strictly confidentially and anonymously. If you want to receive the final report about the survey findings, please specify your email at the end of the questionnaire. If you have any question concerning our study, please contact Dr. Giovanni Randazzo <giovanni.randazzo@lu.unisi.ch>

Thanks for your collaboration!

The EPOCH Team at HOC-Politecnico di Milano and TEC-Lab University of Lugano

SECTION 0: YOUR "CONTEXT"

Date of compilation: day…… month…….. year……..

Name of the company/institution:……

Size of the company/institution (number of people):…….

Your division/department profile (if applicable, please check one or more):
- ❑ Administration
- ❑ Preservation and safeguarding
- ❑ Research
- ❑ Education
- ❑ Technology
- ❑ Other (please specify): ……..

Your company website (if any): http://…….

SECTION 1: CURRENT PRACTICE

The purpose of this section is to explore the methodological approaches you use today in the design process of cultural multimedia applications.

1. How do you design your applications?
- ❑ Informally
- ❑ Formally

In both cases, please describe how:
……
……
……
……

2. Do you use UML?
- ❑ Yes
- ❑ No

If yes, which modeling features of UML do you use most?
……
……

……
……

3. How do you use your design “specifications” (either formal/semi-formal or informal)?
- ❑ To discuss with all members of the design team
- ❑ To discuss with the customer
- ❑ To provide input to the following development activities
- ❑ To produce the required project documentation
- ❑ Other (please specify): ……

4. How do you manage your design process?
- ❑ Informally
- ❑ Formally

In both cases, please describe how:
……
……
……
……

5. Do you use support tool(s) to write/sketch/draw/exemplify your design solutions?
- ❑ Yes
- ❑ No

If yes, please describe which one(s):
……
……
……
……

6. At what degree are your design specifications usually reflected in the final product?
- ❑ 0 %
- ❑ 25 %
- ❑ 50 %
- ❑ 75 %
- ❑ 100%

SECTION 2: THE "NEXT METHODOLOGY"

The purpose of this section is to investigate your opinions about factors and motivations that may prevent or promote the adoption of a new design method.

7. Which improvements and benefits do you expect from a new design method? Please rate your answer.

Aspect	Minimal Improvement	High Improvement
❑ Productivity of development		
❑ Quality of the application		
❑ Usability of the application		
❑ Communication skills within the development team		
❑ Reduction of errors		
❑ Reduction of changes		
❑ Other (please specify): …….....		

Comments and suggestions:
……
……
……

8. If a design method was proposed to you, which characteristic of such a method would you consider most relevant? Please rate your answer.

Characteristic	Absolutely necessary	Very relevant	Relevant	Not relevant
❑ Lightweight (easiness to learn and to apply)				
❑ Standard				
❑ Guidelines and design patterns				
❑ Process model				
❑ *Flexibility* (possibility of using the method in multiple ways)				
❑ *Customizability* (possibility of creating your own version of the method)				
❑ *Scalability* (possibility of adopting a method partially and progressively, first adopting some features only and later extending the use)				

❑ Other (please specify): ……..……..				

Comments and suggestions:
……
……
……

9. If tools for design were proposed to you, which feature would you consider most relevant? Please rate your answer.

Tool feature	Absolutely necessary	Very relevant	Relevant	Not relevant
❑ Appropriate documentation				
❑ Training support				
❑ Consultancy support				
❑ Tools for project management				
❑ Tools for authoring design specifications				
❑ Tools for fast prototyping				
❑ Tools for application generation				
❑ Other (please specify): ……..……..				

Comments and suggestions:
……
……
……

10. How much time do you expect to spend in order to learn and be able to apply a new design method?
- ❑ Less than 1 week
- ❑ 1 to 2 weeks
- ❑ more than 2 weeks

Comments and suggestions:
……
……
……

11. Which type of training do you find more appropriate in order to learn a design method (e.g. on-line courses, mentoring, practice courses)?

……
……
……

GENERAL COMMENTS TO THIS QUESTIONNARIE

……
……
……
……
……
……

Would you like to receive the final report of this study? □ YES □ NO

If yes, please specify your e-mail address:……..

6.2 Questionnaire "Epoch survey on usability evaluation for cultural heritage applications

Questionnaire introduction

This survey is carried on in the context of the EPOCH project (http://www.epoch-net.org/) - workpackage 4.2.

EPOCH, funded by the European Commission under the Community's Sixth Framework Programme. (contract no. IST-2002-507382), is a **network of about a hundred European cultural institutions** joining their efforts to improve the quality and effectiveness of the use of Information and Communication Technology for Cultural Heritage (CH).

Our survey focuses on usability – a fundamental factor for the overall quality of an interactive application. The following questionnaire aims at to investigating the **current usage of usability evaluation methods**, identifying the most common approaches and practices currently adopted by **cultural institutions** in the development of digital applications. We also attempt to identify some critical **requirements for CH evaluation methods**, by understanding the **needs and expectations** of the various stakeholders involved in the evaluation activities of CH applications (e.g., application analysts, domain experts, project managers, information architects, interface designers, end users). In particular, we want to explore the characteristics that should be provided by an evaluation method in order to be *accepted* and *effectively used* in CH projects.

If you are involved in any evaluation activity within a CH project, please help us in this research!

The information you will provide – together with the data collected from other questionnaires - will work pave the ground for defining guidelines, heuristics and best practices concerning evaluation of cultural multimedia, and for the identification for the criteria that a usability methodology should satisfy in order to be **accepted** and effectively **adopted** in the cultural heritage world.

The data collected by our study will be treated strictly confidentially and anonymously. If you want to receive the final report about the survey findings, please specify your email at the end of the questionnaire. If you have any question concerning our study, please contact Dr. Giovanni Randazzo <giovanni.randazzo@lu.unisi.ch>

Thanks for your collaboration!

The EPOCH Team at HOC-Politecnico di Milano and TEC-Lab University of Lugano

SECTION 0: YOUR "CONTEXT"

Date of compilation: day…… month…….. year……..

Name of the company/institution:……

Size of the company/institution (number of people):…….

Your division/department profile (if applicable, please check one or more):

- ❑ Administration
- ❑ Preservation and safeguarding
- ❑ Research
- ❑ Education
- ❑ Technology
- ❑ Other (please specify): ……..

Your company website (if any): http://…….

SECTION 1: CURRENT USABILITY PRACTICE

The purpose of this section is to explore the tools and techniques you use for evaluating cultural multimedia applications.

1. When do you use techniques or methods to evaluate the usability of your interactive applications?

- ❑ Never
- ❑ Sometimes
- ❑ Always

If you answered "never" please go to Section 2.

2. In which phase(s) of a cultural multimedia application lifecycle do you usually perform usability evaluations (e.g. requirements management, design, implementation)?

……
……
……
……

3. Which general approach do you primary used for performing usability evaluations of your applications (e.g. expert analysis, empirical evaluation)?

……
……
……
……

4. If the evaluation of your application is performed through observation of user sessions, where do you collect data?

- ❑ In the "natural" context" of use of the system (e.g. in the museum, in the working environment, at home)
- ❑ In an artificial environment (e.g. usability laboratory)

5. Do you use any specific technique or method for the usability evaluation?

- ❑ Yes
- ❑ No

If yes, please describe which one:

……
……
……
……

6. Who is the typical target of the usability evaluation (e.g. the project manager, the visual designer)?

……
……
……

7. Do you use any software tool or equipment to support the usability evaluation?

- ❑ Yes
- ❑ No

If yes, please describe which one:

……
……
……
……

SECTION 2: THE "NEXT" METHODOLOGY

The purpose of this section is to investigate your opinions about factors and motivations that may prevent or promote the adoption of a new usability evaluation method.

8. How relevant is usability? Please rate your answer.

Aspect	Very relevant	Relevant	Not relevant
❑ Overall			
❑ Contents			
❑ Navigation			
❑ Lay-out			
❑ Services			
❑ Other (please specify): …….			

Comments and suggestions:
……
……
……

9. Which improvements and benefits do you expect from a new usability evaluation method? Please rate your answer.

Aspect	Minimal Improvement	High Improvement
❑ Productivity of development		
❑ Quality of the application		
❑ Usability of the application		
❑ Communication skills within the development team		
❑ Reduction of errors		
❑ Reduction of changes		
❑ Other (please specify): ……..... ...		

Comments and suggestions:
……
……
……

10. If a usability evaluation method was proposed to you, which characteristic of such a method would you consider most relevant? Please rate your answer.

Characteristic	Absolutely necessary	Very relevant	Relevant	Not relevant
❑ Lightweight (easiness to learn and to apply)				
❑ Standard				
❑ Guidelines and heuristics				
❑ Process model				
❑ *Flexibility* (possibility of using the method in multiple ways)				
❑ *Customizability* (possibility of creating your own version of the method)				
❑ *Scalability* (possibility of adopting a method partially and progressively, first adopting some features only and later extending the use)				
❑ Other (please specify): ……..……..				

Comments and suggestions:
……
……
……

11. If tools for usability evaluation were proposed to you, which feature would you consider most relevant? Please rate your answer.

Tool feature	Absolutely necessary	Very relevant	Relevant	Not relevant
❑ Appropriate documentation				
❑ Training support				
❑ Consultancy support				
❑ Method-specific tools for collecting evaluation data				
❑ Method-specific tools for structuring and analyzing the evaluation results				
❑ Support tools for reporting the evaluation results				
❑ Other (please specify): ……..……..				

Comments and suggestions:
……
……
……

12. How much time do you expect to spend in order to learn and be able to apply a new usability evaluation method?
- ❑ Less than 1 week
- ❑ 1 to 2 weeks
- ❑ more than 2 weeks

Comments and suggestions:
……
……
……

13. Which type of training do you find more appropriate in order to learn a usability evaluation method (e.g. on-line courses, mentoring, practice courses)?
……
……
……

SECTION 3: ACCESSIBILITY

The purpose of this section is to investigate your opinions and needs about the factors that contribute to the accessibility of a CH application.

14. Are you aware about accessibility issues and problems in interactive applications?
- ❑ Yes
- ❑ No

15. Do you know the Web accessibility guidelines provided by W3C?
- ❑ Yes
- ❑ No

16. Are you updated about the current research trends in accessibility for interactive applications?
- ❑ Yes
- ❑ No

17. Did you ever developed any application considering accessibility issues?

- ❑ Yes
- ❑ No

If yes, please describe the case(s):

……
……
……
……

18. Are you going to develop any application considering accessibility issues?

- ❑ Yes
- ❑ No

If yes, please describe the case(s):

……
……
……
……

GENERAL COMMENTS TO THIS QUESTIONNARIE

……
……
……
……
……
……

Would you like to receive the final report of this study? □ YES □ NO

If yes, please specify your e-mail address:……..

7. REFERENCES AND BIBLIOGRAPHY

- Ackermann, A.F., Buchwald, L.S., and Lewski, F.H., *Software inspections: An effective verification process*, IEEE Software 6,3 pp. 31-36, May 1989
- Andersen, P.B., A Theory of Computer Semiotics, Cambridge University Press, 1997.
- Blackmon, M.H., Polson, P.G., Kitajima, M., & Lewis, C., *Cognitive Walkthrough for the web*, in *CHI 2002 Conference on Human Factors in Computing Systems*. ACM Press 2002
- Bolchini D, Paolini P, Randazzo G., *Adding Hypermedia Requirements to Goal-Driven Analysis*, In Proc. of the Conference Requirements Engineering 2003.
- Bolchini D., Di Blas N., Paolini P., Triacca L., *Wish you were usable! How to improve the quality of a Museum Web site*, EVA03 Proceedings, March 2003
- Bolchini D., Mylopoulos J., *From Task-Oriented to Goal-Oriented Web Requirements Analysis*, In Proc. of the Conference Web-Wise Sustaining Digital Resources 2003.
- Bolchini, D., Paolini, P. (2004). Goal-driven requirements analysis for hypermedia-intensive Web applications. Requirements Engineering Journal, Springer, RE03 Special Issue 9 2004: 85-103.
- Bolchini, D., Paolini, P., IDM: Interactive Dialogue Model: towards a dialogue-based approach to design interactive hypermedia applications, Deliverable D3 – WED (Web As Dialogue), FNSRS 105211-102061/, May 2004.
- Bolchini, D., Paolini, P., *Interactive Dialogue Model: a Design Technique for Multi-Channel Applications*, submitted to IEEE Transaction on Multimedia, July 2004.
- Brinck, T., Gergle, D., Wood, S.D, *Usability for the Web: Designing Websites that Work*. San Francisco, San Diego, New York, Boston, London, Sydney, Tokyo: Morgan-Kaufmann, 2002.
- Brinck, T., Gergle, D., Wood, S.D., *Usability for the web*, Morgan Kaufmann, 2002
- Cantoni, L. & Paolini, P. *Hypermedia Analysis: Some Insights from Semiotics and Ancient Rhetoric*, Studies in Communication Sciences vol. 1, 2001, pp 33-53.
- Carroll, J., *Making Use – Scenario-based design of Human-Computer Interactions*, MIT Press, 2002
- Cato, J., *User-Centred web Design*, Addison Wesley, 2001
- Ceri, S., Fraternali, P., Bongio, A. et al., Designing Data-intensive Web Applications, Morgan Kaufmann, 2002.
- Chi, E. H., Pirolli P. and Pitkow J., *The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a website*, In Proceedings of the Conference on Human Factors in Computing Systems, pp. 161- 168, The Hague, The Netherlands, April 2000, New York, NY: ACM Press
- Chi, E.H., Adam S. Rosien, A.S. and Heer, J., *LumberJack: Intelligent Discovery and Analysis of web User Traffic Composition*. In Proc. ACM-SIGKDD Workshop on web Mining for Usage Patterns and User Profiles (webKDD 2002), ACM Press, July 2002. Edmonton, Canada
- Conallen, J., Building Web Applications with UML (second edition), Addison-Wesley, 2003.
- Costagliola, G., Ferrucci, F. & Francese, R. (2002). Web engineering: Models and methodologies for the design of hypermedia applications. Handbook of Software Engineering & Knowledge Engineering
- De Troyer, O., Leune, C., WSDM: a User-Centered Design Method for Web Sites, in Proceedings 7th International Wolrd Wide Web Conference, Brisbane, 1997.

- DELOS – European Research Network of Excellence on Digital Libraries, (Contract N. IST-2002-507618), funded by the European Union, http://delos-noe.iei.pi.cnr.it.
- Di Blas N., Guermandi M.P. Orsini C., Paolini P., *Evaluating The Features of Museum websites*, Museum & web Conference, Boston, April 2002
- Di Blas, N. & Paolini, P. *"There And Back Again": What Happens to Phoric Elements in a "Web Dialogue"*. Journal of Document Design, 4(3), 2003, pp 194-206.
- Di Blas, N. & Paolini, P. *Do we speak with computers? How linguistics meddle with the Web*, In E. Rigotti, A. Giacalone-Ramat & A. Rocci (Eds.). Linguistics & new professions. Milano: Franco Angeli, 2003, pp 221-233.
- Di Blas, N. & Paolini, P. *La Retorica e il Web*, in D'Alessandro, P. (ed.), Internet e la filosofia, LED, Milano, 2001, pp 39-49.
- Di Blas, N. *Deixis & Formulations – When the Web speaks of himself*. SCOMS – Studies in COMmunication Sciences, 2003.
- Di Blas, N., Paolini, P., M. Speroni, A. Bienert, *Listen to a Website: accessibility (beyond current standards) and a market opportunity*, ICHIM 2003 proceedings, Paris, 2003.
- Di Blas, N., Paolini, P., Speroni M., Capodieci, A., *Enhancing accessibility for visually impaired users: the Munch's exhibition*. In Bearman D. & Trant J. (eds), Museums and the Web 2004. Selected Papers from an International Conference, Archives & Museum Informatics, Arlington, Washington, U.S.A., 2004.
- Di Blas, Paolini, Speroni (2004). Web Accessibility for Blind Users Towards Advanced Guidelines. UI4ALL Conference (User Interfaces for All), Wien.
- Di Blas, Paolini, Speroni, Capodieci (2004). Enhancing accessibility for visually impaired users: the Munch's exhibition. Museums and the Web 2004 Conference, Arlington, USA.
- Dix, A., Finlay, J., Abowd, G., and Beale, R., *Human Computer Interaction*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1998
- Dix, A., Finlay, J., Abowd, G., Beale, R., Human-Computer Interaction, Prentice Hall, 2002 (second edition).
- Donahue G.M, Weinschenk S., Nowicki J., *Usability Is Good Business*, Compuware Corporation Report, July 1999
- Dumas J.S., Redish J.C., *A practical Guide to Usability Testing*, Intellect 1999
- EPOCH – European Research Network in Processing Open Cultural Heritage (Contract N. IST-2002-507382), funded by the European Union, http://www.epoch-net.org.
- Garzotto F., Paolini P. (1993) HDM- A Model-Based Approach to Hypertext Application Design. In ACM Transactions on Information Systems, Vol. 11, No1 January p1-26.
- Garzotto, F., Matera, M., Paolini, P., *To Use or Not to Use? Evaluating Usability of Museum Web Sites*, Museums and the Web 1998. Selected Papers from an International Conference, Archives & Museum Informatics, Toronto, CA
- Gilb, T., *Principles of Software Engineering Management*, Addison Wesley, Reading, MA, 1998
- Gómez, J., Cachero, C., Pastor, O., Conceptual Modeling of Device-Independent Web Applications, IEEE Multimedia, April-June 2001 (Vol. 8, No. 2).
- Green, T.R.G. & Benyon, D.R., *The skull beneath the skin; Entity-relationship modeling of information artifacts*, int. J. Hum.-Comput. Stud., vol. 44, no. 6, pp. 801-828, June, 1996
- Hewett, T., Baecker, R.M. et al., Dialogue Techniques, ACM SIGCHI Curricula for Human-Computer Interaction, ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group, available at: http://sigchi.org/cdg/cdg2.html.

- Isakowitz T, Stohr EA., Balasubramanian P. (1995) RMM: A design Methodology for Structured Hypermedia Design. In Communications of the ACM Vol.38 No8 August pp 34-44.
- Ivory, M., Hearst, M., *Statistical Profiles of Highly-Related web Sites*, In Proceedings of CHI Conference, Volume 4, Issue No. 1, pp. 367-374, 2002
- Jeffries, R., Miller, J., Wharton, C., and Uyeda, K.M., *User interface evaluation in the real world: A comparison of four techniques*, in Proc. ACM CHI, 1991, pp. 119-124.
- Kitajima, M., Blackmon, M.H., & Polson, P.G. (2000) *A comprehension-based model of web navigation and its application to web usability analysis*, in S. McDonald, Y. Waern & G. Cockton (Eds.), *People and Computers XIV – Usability or Else!* (*Proceedings of HCI 2000*, pp. 357-373)
- Koch, N. (2001) A Comparative Study of Methods for Hypermedia Development. Technical Report 9905, Ludwig-Maximilians-Universitaet
- Krug S., *Don't make me think. A Common Sense Approach to web Usability*, Que 2000
- Lange DB. (1996) An Object-Oriented design Method for Hypermedia Information Systems. In Journal of Organizational Computing & Electronic Commerce, Vol. 6 (3).
- Lim, K.H., Benbasat, I. and Todd, P.A., *An experimental investigation of the interactive effects of interface style, instructions, and task familiarity on user performance*, ACM Trans. Comput. Hum. Interact., vol. 3, no 1,pp. 1-37, January 199
- Linell, P., & Luckmann, T. *Asymmetries in Dialogue: Some Conceptual Preliminaries.* In I. Markova & K. Foppa (Eds.), Asymmetries in Dialogue Harvester Wheatsheaf: Hemel Hempstead, 1991, pp. 1-20.
- Madsen, K.H., *Special issue on the diversity of usability practices*, Commun. ACM, vol. 42, no. 5, 1999
- MAIS project (Multichannel Adaptive Information Systems), project funded by the Italian Ministry of Education and Research (MIUR), http://black.elet.polimi.it/mais/index.php.
- Matera, M., Costable M.F., Garzotto F., Paolini P., *SUE Inspection: An Effective Method for Systematic Usability Evaluation of Hypermedia*, IEEE Transaction, Vol.32, No. 1, January 2002
- Nanard, M., Nanard, J., King, P., IUHM, A Hypermedia-Based Model for Integrating Open Services, Data and Metadata, in Proc. Hypertext '03 Conference, 2003.
- Nevile, L. & McCathieNevile, C., *The Virtual Ramp to the Equivalent Experience in the Virtual Museum. Accessibility to Museums on the Web*. In D. Bearman & J. Trant (Eds.). Museums and the Web 2002. Selected Papers from an International Conference. Pittsburgh: Archives & Museum Informatics, 2002, pp 93-99.
- Nielsen, J., *Designing web Usability*, New Riders, 1999
- Nielsen, J., Mack, R., *Usability Inspection Methods*, Wiley 1994
- Paolini P., Garzotto F., *Design patterns for WWW hypermedia: problems and proposals*, in Proc. ACM HT '99, Workshop on "Hypermedia Development: Design patterns in Hypermedia", Darmstadt, 1999.
- Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992) *Cognitive walkthroughs: A method for theory-based evaluation of user interfaces*, in *International Journal of Man-Machine Studies*, 36, 741-773
- Ratner, J., Grose, E.M. and Forsythe, C., *Characterization and assessment of HTML style guides, In Proceedings of the Conference on Human Factors in Computing Systems*, volume 2, pp. 115-116, Vancouver, Canada, April 1996, New York, NY: ACM Press
- Rocci, A., Paolini P., Towards a model for the analysis and annotation of information dialogues, Deliverable D1 – WED (Web As Dialogue), FNSRS 105211-102061/, May 2004.

- Roethlisberger, F.J. and Dickson, W.J. *Management and the Worker*, Boston, Mass., Harvard University Press, 1939
- Rossi G, Schwabe D, Guimaraes R. (2001) Designing Personalized Web Applications. In proceedings WWW10, Hong-Kong, May.
- Rosson, M.B., Carroll, J., *Usability Engineering*, Morgan Kaufmann, 2002
- Schwabe D., Rossi G., *An Object Oriented Approach to Web-Based Application Design, Theory and Practice of Object Systems*, J. Wiley, 1998.
- TEC-Lab – Technology Enhanced Communication Laboratory, *Web as Dialogue: Interpreting Navigational Artifacts as Dialogic Structures*, internal report, 2002.
- Theofanos, M.F. & Redish, J., *Guidelines for Accessible – and Usable – Web Sites: Observing Users Who Work With Screenreaders*. Interactions, vol. X.6, pp 36-51, 2003.
- Triacca, L., *Wish you were usable! How to improve the quality of a Museum Web site*, EVA 2003, Florence, Italy 2003
- UWA Consortium, *D7: Hypermedia and Operation Design*, Deliverable Project, 2001, http://www.uwaproject.org.
- UWA consortium, Hypermedia and Operation Design: Model, Notation, and Tool Architecture, UWA Project, IST-2000-25131, Deliverable D7, <www.uwaproject.org>, 2001.
- VNET5 Consortium, *Best Practice Manual for User Centred Product Creation in Interactive Electronic Publishing*, June 2002
- VNET5 Consortium, *Common Approach to user centered product creation*, Deliverable 2.1.1, www.vnet5.org, October 2001
- Whiteside J., Bennet J., and Holtzblatt K., *Usability engineering: Our experience and evolution*, in Handbook of Human-Computer Interaction, M.Helander, Ed. Amsterdam, The Netherlands, North-Holland, 1988, pp.791-817
- Withrow, J., Brinck, T., Speredelozzi, A., *Comparative Usability Evaluation for an e-Government Portal*, Diamond Bullet Design Report 1-00-2, Ann Arbor, MI. Dec. 2000.
- Woukeu, A., Carr, L., Wills, G. and Hall, W. (2003) Rethinking Web Design Models: Requirements for Addressing the Content. Technical Report ECSTR-IAM03 -002, Department of Electronics and Comnputer Science, University of Southampton.

Online resources:

http://www.w3.org/WAI/: Web Accessibility Initiative
http://www.w3.org/2001/di/: Device Independence Activity
http://www.w3.org/2002/mmi/: Multimodal Interaction Activity
http://www.w3.org/Voice/:"Voice Browser Activity - Voice enabling the Web!
http://www.minervaeurope.org/publications/qualitycriteria.htm