Abstract Tasks: a Tool for the Inspection of Web Sites and Off-line Hypermedia

F. Garzotto, M. Matera, P. Paolini

HOC – Hypermedia Open Center
Dipartimento di Elettronica, Politecnico di Milano
P. zza Leonardo da Vinci 32, 20133 Milan, Italy
E-mail: [garzotto, matera, paolini]@elet.polimi.it

ABSTRACT

This paper discusses a systematic method for inspecting the usability of on-line and off-line hypermedia. The core idea of our approach is the use of an organized list of Abstract Tasks to guide the inspector's activity. An Abstract Task specifies a pattern of inspection operations that the evaluator is required to perform on some specific features of a hypermedia. Abstract Tasks capture our expertise in usability inspection, and express it in a precise and understandable form, so that it can be easily "reproduced", communicated, and exploited. They help transferring usability expertise from experienced to inexperienced inspectors, and sharing know-how among different evaluators. Thus, different inspectors who systematically apply the same set of Abstract Tasks are more likely to come up with consistent results, and the overall quality of their inspection (in terms of completeness and accurateness of the findings) is greatly improved.

The paper briefly introduces the background of our approach and explains the rationale of Abstract Tasks. It also provides some examples of Abstract Tasks (out of the 43 currently defined) and of inspection results achieved by applying them to inspect WWW sites and commercial CD-ROMs.

KEYWORDS: Usability Inspection, Evaluation Methodology, Hypermedia Quality.

INTRODUCTION

Most of the reported experiences on usability evaluation of hypermedia [Bev97, Bot92, Far96, UIE97, Yam95] adopt empirical techniques based on user testing, where panels of users are asked to perform some kind of actions, while their behavior and reactions are monitored.

Hypertext 99 Darmstadt Germany

The main disadvantages of inspection techniques over empirical testing are the great subjectiveness of the evaluation (different inspectors may produce different outcomes) and the heavy dependence upon the skills of the inspector. The main advantage is the cost-saving: while user experiments are expensive, inspection-based evaluation "save users" [Jef 91, Nie93, Nie94], and do not require special equipment, nor lab facilities. For this reason, inspection methods have seen an increasing widespread use in these years, especially in industrial environments.

The literature [Nie94] reports a variety of experiences of usability inspection, evaluating general interface features that are common to the majority of interactive systems, but few of them focus on hypermedia specific aspects. Furthermore, all these works are largely based on the individual know-how of inspectors, on their skills, and on their personal judgement. The inspector experience, hardly repeatable by others, have not led so far to the definition of standardized processes nor to precise guidelines. As a matter of fact, inspection is largely conceived as a kind of "art" - a largely subjective process.

Our research investigates the use of inspection techniques for evaluating the usability of hypermedia specific features. We abstract from the application domain of a system or the intended user tasks, and rather focus on the design strength of the application, inspecting its information structures, paradigms for navigation, ways of interacting with multiple active media. We try to address questions that normally arise when one analyses a hypermedia for usability, such as: "what should I look for?", "Where should I start from?", "Which elements are more critical and worth to invest more time?". Our ultimate goal is to make usability evaluation systematic and efficient, and to encourage standardization across different evaluators and evaluation processes.

We have defined a set of operational guidelines to answer the above questions, organizing the inspection process into groups of well defined activities, each one focusing on a different aspect of hypermedia. Such guidelines are called Abstract Tasks, and are the main topic discussed in this paper. The following section will explain the concept of Abstract Tasks and its rationale. Then we will present examples of Abstract Tasks and discuss how they can be used in practice, reporting fragments of inspection results achieved by applying them to WWW sites and off-line hypermedia (commercial CD-ROMs). In the last section, we discuss some critical issues related to the inspection process, to the validation of our work, and to the future directions of our research.

ABSTRACT TASKS: BACKGROUND AND RATIONALE

An Abstract Task describes types of "actions" that the inspector is required to perform on some hypermedia specific features of the product under evaluation. Our experience, based on our individual work and on validation experiments we are carrying out, is that different inspectors, applying systematically the same set of Abstract Tasks, come up with more consistent and comparable results, and in a shorter time than without using our approach.

The idea of Abstract Tasks was developed while we were given the goal of helping the industrial consortium CORINTO[1] to set up an evaluation laboratory [Err96], and to support the usability evaluation of a large number of hypermedia applications, either produced inside the consortium, or taken from the market. One of the requirements of this project was that evaluations should have been based both upon empirical testing and inspection, with inspection to be applied more intensively, and empirical testing confined to few special cases (given the complexity and costs of setting up experiments).

Within this project, we developed a usability evaluation methodology named SUE (Systematic Usability Evaluation), in co-operation with a group of researcher at University of Lecce and University of Bari. SUE is not hypermedia specific, but provides general evaluation framework for interactive systems that must be specialized for any specific category. A core idea of SUE is that usability evaluation should address a variety of aspects of a product, analyzing it under different perspectives. Some of these aspects refer to general lay-out features common to all interactive systems, others are more specific to a particular type of product, or a particular domain of use. The process of analyzing different perspectives may require different activities and judgement criteria. Another key concept of SUE is that the most reliable evaluation results can be achieved by systematically combining inspection with user testing [Jef91]. The inspection is carried out first; then user testing is used to validate and refine the result of the inspection, and is carried out by requiring end-users to perform some "concrete" tasks, i.e., specific activities, defined on the basis of the results of the inspection. The use of concrete tasks makes the empirical testing better organized and cost-effective. The combination of inspection and empirical testing ensures the most accurate evaluation results, coupled with cost-effectiveness. Finally, SUE is model-based since models are used to precisely shape up the evaluation activities. The hypermedia model we use is HDM (Hypermedia Design Model), developed at Politecnico di Milano.

In defining a systematic method for inspection, we were given the following requirements:

- to obtain high quality evaluations, even when using young, relatively, inexperienced inspectors;
- to obtain consistent evaluations across a certain number of different inspectors;
- to obtain a good degree of "precision" (most of the reported problems had to be actual problems) and "recall" (most of the actual problems had to be reported);
- from evaluations of internally produced applications, to obtain precise feedback for improving the applications, not just a good/bad score.
- to support sharing of usability evaluation know-how within evaluators, possibly of different levels of expertise.

The notion of Abstract Tasks was developed and refined to tackle the above requirements.

Essentially, Abstract Tasks capture our evaluation expertise. Within our research group, the availability of a large collection of CD-ROM's and "local" web sites (i.e. sites downloaded on CD-ROM for an efficient browsing) has led to a great activity of "reading" and "learning" from other people work. We have gradually developed a small group of "expert" inspectors, able to detect the main navigational features of an application fast and quite reliable[2].

Since our laboratory also supports a large teaching activity (more than 200 students, each year, in different classes), we had to teach inexperienced people how to "read" and evaluate hypermedia applications, both on CD-ROM and on WWW. Trying to explain our method of analysis, slowly we began to develop a number of "rules" of what to do and what to look for, while performing an evaluation. This is how Abstract Tasks started to shape up: capturing the actions of expert inspectors, and trying to explain them to inexperienced readers. These rules, expressed in a rigorous way, became the ancestors of Abstract Tasks.

The technical problem was how to describe inspection operations or actions, without referring to a specific application, but in general terms. Modeling was the solution: a generic application could have been described using the primitives, the concepts and the terminology of HDM [Gar93, Gar95].

The actions to be performed during the evaluation were therefore framed using a generic HDM schema. During an inspection, two were the tasks for the inspector:

a) to develop a viable HDM schema for the application (if the schema was not already available);
b) to apply the Abstract Tasks on the basis of the schema.

For internally developed applications, HDM was used as part of the design methodology, and therefore task "a)" was unneeded. For external applications, the HDM model was not available, but had to be developed; therefore, task

[1] CORINTO (COnsorzio RIcerca Nazionale Tecnologia ad Oggetti) is an Italian consortium involving IBM-SEMEA, APPLE-Italy, SELFIN. Its main purpose is to investigate the impact of object oriented and multimedia technologies upon the application development market.

[2] A few times it happened to us that we "clarified " to the authors of commercial applications what the actual behavior of those applications really was, highlighting strange features that they had never noticed.

"a)" required a specific care. In this paper, however, we focus upon what Abstract Tasks are, and how they are intended to be used.

An Abstract Task (AT for short), in essence, requires the inspector to perform some standard operations, also specifying what to look for. An AT is "abstract", in the sense that the operations are formulated independently from a particular application, referring to categories of application constituents. Its structure is the following:

Focus of Action: it specifies which part of the application is mainly involved.

Activity Description: it precisely describes the activity to be performed.

Intent: it explains the purpose of the AT, trying to make clear which is the specific goal to be achieved.

Output: it describes what should be the output of the inspection performed by applying the AT.

Optionally, a ***Comment*** can be also provided.

The key part of an AT is obviously the "activity description", where what must be done is specified. The generic activity is precisely described in terms of actions on various object types, but its translation into an action upon specific objects of a concrete application sometimes can be non trivial. The first job is to correctly identify all the different parts of the application that should be tested; secondly, it is needed to correctly perform the action specified; third, the outcome of the action must be properly interpreted.

The original list of ATs included 23 entries, the current list [Gar98] includes 43 entries. ATs are organized into a number of categories and sub-categories, depending on which hypermedia feature they focus on:

- *Content ATs* (15 elements)

These ATs concern content structures in-the-small, i.e., atomic information elements ("slots", in HDM terminology). Content ATs are organized in two sub-categories:

- *Active slots ATs* (8 elements)
  They concern the behavior of time-based slots (video, audio, animations, etc.). Their goal is to evaluate the effectiveness of the interaction with a single active slot or a group of active slots, to check the consistency and regularity of the interaction mechanisms across different slots, and to analyse the "interplay" between active slots and navigation (i.e. how they can affect each other).
- *Passive slots ATs* (7 elements)
  They address the usability of interaction with time-independent slots (text, data, still images, etc).

- *Structure ATs* (6 elements)

These ATs focus on the content organization in-the-large. They are organized in two sub-categories:

- *Hyperbase Structure ATs* (3 elements)
  They consider the structures representing the application domain contents (hyperbase, in the HDM terminology). They aim at evaluating if the complexity of the various structures is adequate for representing the underlying information content.
- *Access Structure ATs* (3 elements)
  They focus on the organization of structures for accessing the hyperbase content ("collections" in HDM, such as "guided tours").

- *Navigation ATs* (20 elements)

These ATs focus on the navigational aspects of the application, and are organized in two sub-categories:

- *Hyperbase Navigation ATs* (6 ATs), inspecting the paradigms to explore the application domain content.
- *Access Navigation ATs* (14 ATs), inspecting the paradigms to explore indexes and guided tours.

- *Miscellany ATs* (2 elements), which address usability issues related to reuse of structures in different contexts of the application [Gar96].

The list of ATs is still under evolution. Our perception is that, at the end of the process of discovering and specifying ATs, we will get a fairly large set - approximately 50 ATs, with about 50% of them representing the basic set, and with the rest being considered as "advanced features".

The following section will describe few examples of ATs defined so far, with some excerpts of findings when ATs were applied to different commercial applications on CD-ROM and WWW sites.

EXAMPLES OF ABSTRACT TASKS AND INSPECTION RESULTS

AS-5[3]: "Navigational Behavior of Active Slots"

Focus of Action: an active slot + links

Activity Description: consider an active slot:

1. activate it, and then follow one (or more) link(s) *while the slot is still active*; return to the node where the slot has been activated, and verify the actual slot state;
2. activate the active slot; *suspend it*; follow one (or more) link(s); return to the node where the slot has been suspended and verify the actual slot state;
3. execute 1 and 2 traversing different types of links, both to leave the original node and to return to it;
4. execute activities 1 and 2 by using only backtracking to return to the original node.

Intent: to evaluate the cross effects of navigation on the behavior of active slots.

Output:

A. a description of the behavior of the active slot, when its activation is followed by the execution of navigational links and, eventually, backtracking.
B. a list and short description of possible unpredictable effects or semantic conflicts generated in the source and/or in the destination node.

3 AS-5 means "the fifth AT for Active Slots". Active slots, in HDM, are those values (such as video, audio or animation) which exhibit a dynamic behavior, i.e. a change of state with time.

Inspection Results by Abstract Task AS-5 in "Le Louvre" CD-ROM

In the CD ROM *"Le Louvre"* [Reu94], for each painting different types of nodes supply different types of information. In particular, a node of type "Loupe" (see Figure 1) provides an analysis of the painting based on a progressive "zoom" of details, and contains an animation slot synchronized with an audio comment.

If a link is selected while animation and audio are still active, the audio comment continues till the end of the current audio "slice", although the current node is immediately replaced by the link destination node. Also, any further click does not interrupt the audio slice, as it generally happens for the other audio comments in the application. Therefore, users find themselves on a content which has nothing to do with the comment they are listening to, and they are not able to deactivate it.

Figure 1: Node of type "Loupe" in the CD-ROM "Le Louvre"

HB-N4[4]: "Complexity of applicative[5] navigation patterns"

Focus of Action: an applicative link.

Activity Description: select an applicative link:

1. navigate from the source node to one of the target nodes;
2. randomly visit one of the target nodes;
3. move directly from a target node to another target node;
4. systematically visit all the target nodes;
5. from target nodes try to navigate to the source node, without using backtracking commands.

Intent: given an applicative link, to evaluate:

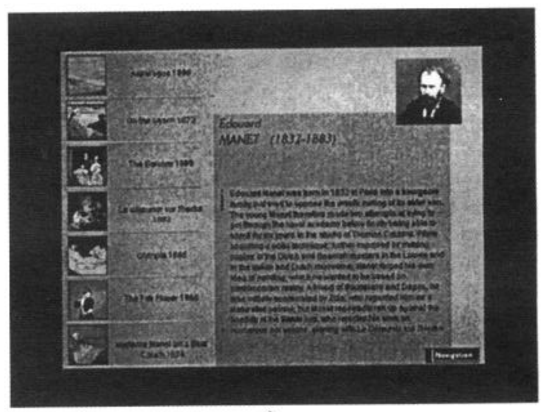
A. its navigation pattern;

B. its symmetric property, i.e., if the applicative navigation pattern includes the definition of a symmetric link from the target node to the source node.

Output: a description of the navigational pattern for the applicative link under evaluation, and a judgement about its appropriateness, with respect to the semantic relationship represented by the link.

Inspection Results by Abstract Task HB-N4 in the "Musée d'Orsay " CD-ROM

In the CD ROM *Musée d'Orsay* [Reu96], there is an interesting example of applicative navigation pattern. An applicative link is defined between the two entity types "Author" and "Work", with cardinality "one-to-many". The link is symmetric, since from each painting it is possible to navigate to the correspondent author, and vice-versa from the author it is possible to go to his/her paintings. Starting from the author, and following one of the available links (see Figure 2a), it is possible to go to one of the works. From there, through the button "Next from artist" (see Figure 2b), users are able to visit all the other works, without going back to the artist node. This effective navigation pattern is (unfortunately) often missing in most CD-ROM or in WWW applications.



a)

b)

Figure 2: In the CD_ROM "Musée d'Orsay", the source node (Figure a), and one of the target nodes (Figure b) for the applicative link between the entities "Artist" and "Painting".

[4] HB-N4 means "the fourth AT for the Navigation in the Hyperbase". The Hyperbase, consisting of entities and applicative links among them, is the hearth of an HDM application. Entities represent domain objects, and applicative links represent semantic relationships among them.

[5] The word " applicative" is a technical term in HDM; it is applied to links within the hyperbase which denote domain-dependent semantic relationship among information items (as opposite to "structural" links, that express domain-independent topological relationships, or access links within access structures.) Applicative navigation style denote the navigation induced by applicative links.

Inspection Results by Abstract Task HB-N4 in "The National Gallery of Washington" Web Site

An example of one-to-many applicative link in this web site [NGA98] is the one from "Location", describing an area of the museum, to the artworks exhibited in that area. Differently from the "Musée d'Orsay" CD-ROM, no guided-tour navigation among the target nodes is provided, and the only way for reaching each target node is to pass through the node of type "Location" (see Figure 3), and follow the applicative link.

Figure 3: The node of type "Location" in the "National Gallery of Washington" site.

HB-N5: "Consistency of applicative navigation patterns"

Focus of Action: the set of *applicative links of the same type*.

Activity Description: apply Abstract Task HB-N4 on *applicative links of the same type*, and compare their navigation patterns.

Intent: to evaluate the consistency of navigation patterns across applicative links of the same type.

Output: a statement saying if the definition of the applicative links of the same type is based upon the same navigational pattern.

Comment: this Abstract Task may look surprising, at first sight, but consistency across applications is very important for usability and, more than it could be expected, is violated in applications.

HB-N6: "Regularity of applicative navigation patterns".

Focus of Action: the set of *applicative links of different types*.

Activity Description: apply Abstract Task HB-N4 on applicative links of different types, and compare their navigation patterns.

Intent: to evaluate the degree of regularity of navigation patterns across applicative links of different type.

Output: a statement saying if applicative links of different type have the same or a similar navigation patterns, for moving from the source node to the target nodes and vice versa. If the definition of navigation pattern across the application is not regular, the statement should also mention if this design choice is acceptable, or if it affects usability, having impact on the users' orientation and predictability.

Comment: while consistency concern links of the same type, regularity concern links of different types. Links of different types may behave differently, of course; still, for the sake of predictability which is a fundamental usability factor, unnecessary differences in behavior should be avoided.

Inspection Results by Abstract Task HB-N6 in "Musée d'Orsay " CD-ROM

In the "Musée d'Orsay" CD-ROM, all the applicative links defined in the application follow a same navigational pattern, which is the one previously described for the link between the entities "Authors" and "Work". For example, another applicative link between the entities "Room" and "Work" enables the navigation from one room to the works located in the room. From the node of each work is then possible to follow the symmetric link to the room, or to visit sequentially all the other works of the room (i.e., all the target node for the applicative link), by activating the link "Next in room".

AL-N1[6]**:** "Complexity of collection navigation patterns"

Focus of Action: a collection + the set of collection links.

Activity Description: select a collection:

1. from the collection center[7] access a generic collection member;
2. from a generic member[8], access the collection center (if any);
3. from a generic member of the collection, access:
    a) the previous and the next member in the collection order;
    b) the first collection member and the last one;
    c) another arbitrary member;
4. from the first (respectively, the last) member, try to go "previous" (respectively "next");
5. in parallel with each one of the previous activities, every time a member is accessed, verify if it is possible to identify its placement within the collection.

Intent: in a given collection, to evaluate:

A. the collection navigation pattern;
B. the mechanisms supporting the navigation status visibility, i.e., those elements which give indications about the placement of members within the collection.

Output: a description of the collection navigation pattern, and of the mechanism for the navigation status visibility, and a judgement about the way they support the navigation through collection members.

[6] AL-N1 means "the first AT for the Navigation of the Access Layer". The Access Layer, in HDM, consists of Collections of objects. Each collection creates an Access Path, made available as an Index, a Guided Tour or both.

[7] The collection center is the node where the collection starts, e.g. the index page or the starting page of a guided tour.

[8] A collection member is an object (e.g. a "painting") belonging to a collection (e.g. "masterpieces").

Inspection Results by Abstract Task AL-N1 in the "Musei Vaticani" Web Site

In the "Musei Vaticani" Web site [Vat98], the information about the museum masterpieces is organized in several collections, each one corresponding to a real gallery in the museum (Gallery of Tapestries, Gallery of Paintings, etc.). The site has a simple structure, quite intuitive and easy to grasp. Unfortunately, there is a lack of commands for navigating through the collection members: neither "next", nor "back" links are provided. Therefore, for visiting more than one work in a collection, users are forced to a continuous navigation from the list of works (the center of the collection) to a specific one, than back again to the list for choosing another work in the collection.

Inspection Results by Abstract Task AL-N1 in the "National Gallery of Washington" Web Site

The access structures of this site [NGA98] are quite deep, and could be probably hard to explore if it hadn't a relatively rich set of collection links. There are from 3 to 5 levels of nesting in the collections. With exception of the collection "Exhibitions", the links in a collection enable the navigation from the center to each member, and vice versa. Within "Exhibitions", "next" and "previous" links are also provided to navigate linearly across members. Furthermore, there are a number of collections named "Guided Tours", which provide circular linear navigation, allowing users to reach the first member from the last one. Horizontal jumps among "sibling" and vertical jumps to "ancestors" are usually not allowed, but the indexes "General Information", "The Collection", "Exhibitions" "Program & Events", "Resources" and "Gallery Shops" can be reached from everywhere in the site.

DISCUSSION AND FUTURE WORK

It is indubitable that inspection of complex hypermedia (or evaluation of hypermedia, in general) is still an "art", in the sense that a great deal is still left to the skills, experience, ability and capability of the inspectors. The aim of our approach, based on the systematic use of Abstract Tasks, is to help usability inspectors share and transfer their evaluation know-how, to make it easier to learn by newcomers, and to achieve more reliable (in terms of quality of results), more consistent (i.e., more independent from the people carrying on the evaluation), and more cost-effective evaluation processes.

So far we have tried to informally validate our method, by asking inexperienced hypermedia readers to apply ATs while inspecting several hypermedia. The results have been quite encouraging, since most of them, after a brief exposition to Abstract Tasks, have been able to improve their reading abilities. Furthermore, we have noticed that different inspectors using the same set of Abstract Tasks come up to similar conclusions. The main differences among experienced inspectors and inexperienced ones seems to be on speed (experienced inspectors are faster) and accuracy of the reporting (experienced inspectors provide more precise feedback).

In order to validate the method in a more formal and objective way, we have planned a controlled experiment, to collect some empirical data about the performance of different groups of evaluators having different profiles (usability expert vs. non expert), and under different conditions (using vs. non using Abstract Tasks). Their evaluation process and the output of their inspection will be observed, measured in terms of time needed for accomplishing an evaluation session, type and number of usability defects discovered during an inspection, and statistically compared. The goal of the experiment is to validate the following assumptions, founded on the experience achieved so far but not documented in a precise way:

- ATs really support the standardization of results, along different evaluators with a different experience. This assumption comes from the fact that Abstract Tasks precisely prescribe which objects to look for in the application, and which evaluation activities to perform on them, in a way completely independent from the evaluator experience in reading and inspecting hypermedia.
- By providing an organized check-list of operations to perform, the use of ATs reduces both the time required for training evaluators, and the time needed by evaluators for conducting an evaluation sessions.

There are a number of problems that we perceive, in the use of the methodology. First of all, Abstract Tasks require that the inspectors understand the HDM model[9]; we cannot circumvent the problem, since a strong modeling is the key point to standardize inspection. An additional problem, perhaps the most crucial one, is that the inspector must have an HDM model of the application at hand , in order to apply ATs. Where does this model come from? If the application was developed with HDM, the model is provided by the designer. Since this is, unfortunately (at least for us), an uncommon situation, the inspector must develop a model of the application while inspecting it. It turns out that this sort of "reverse modeling" is more difficult than applying the ATs. And, in fact, we have reached the conclusion that this is the only difficult part: once the model is developed, everything else is quite straightforward. In order to tackle this problem we have started working on a set of different ATs: those describing actions that allow an inspector to develop a structured model of the hypermedia application under evaluation. The definition of these "reverse modeling" Abstract Tasks is an additional direction of our current research.

Finally, the output of ATs should help to correlate inspection results to (categories of) user tasks. Presently, the inspector analysis of a given feature, which results from the execution of an AT, focuses on the design strength of that feature, and abstracts from the domain and intended goals of the system; the inspector judgement must be later weighted and interpreted in light of the actual context of use. To help this interpretation process, it would be useful to identify some user tasks in which some design solutions are appropriate or not appropriate, mapping design patterns [Gam95, Sch97] discovered with the ATs into situations of use.

ACKNOWLEDGMENTS

This work has been partially founded by the CORINTO

[9] As a matter of fact we actually use ATs to teach the HDM model.

Consortium, the European Commission Project "SITMOON", the MURST project "MIDA", and the CNR Project "Museo Virtuale dell'Informatica". We acknowledge Prof. M. F. Costabile, from Department of Computer Science, University of Bari, Italy, for her valuable suggestions.

REFERENCES

[Bev97] Bevan N. "Usability Issues in Web Site Design". Proc. of HCI International 97 (San Fancisco - CA), 1997.

[Bot92] Botafogo R.A., Rivlin E., Shneiderman B., "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics". *ACM Trans. on Inf. Systems* 10 (2), April 1992.

[Err96] Errore S., Costabile M.F., Matera M. "Design and Prototype Implementation of a Usability Laboratory" (in Italian). CORINTO Technical Report, Bari, Italy, September 1996.

[Far96] Faraday P., Sutcliffe A. "An Empirical Study of Attending and Comprehending Multimedia Presentations". In Proc. ACM Multimedia'96 (Boston, MA), ACM Press, 1996.

[Gam95] Gamma E., Helm R., Johnson R. Vlissedes J. "Design Patterns - Elements of Reusable Object Oriented Software", Addison Wesley, 1995.

[Gar93] Garzotto F., Paolini P., Schwabe D. "HDM - A Model Based Approach to Hypermedia Application Design". *ACM Trans. on Information Systems*, 11 (1), January 1993.

[Gar95] Garzotto F., Mainetti L., Paolini P. "Hypermedia Design, Analysis, and Evaluation Issues". In Comm. ACM, 38 (8), August 1995.

[Gar96] Garzotto F., Mainetti L., Paolini P. "Information Reuse in Hypermedia Applications". In Proc. HT'96 (Boston - MA), ACM Press, March 1996.

[Gar97] Garzotto F., Matera M. "A Systematic Method for Hypermedia Usability Inspection". In *The* New Review of Hypermedia and Multimedia, Vol. 3, 1997.

[Gar98] Garzotto F., Matera M., Paolini P. "Abstract Tasks for Hypermedia Usability Inspection". Dept. Of Electronics and Information, Polytechnic of Milan, Technical Report No. 03-98, 1998.

[Jef91] Jeffries R., Miller J.R., Wharton C., Uyeda K.M. "User interface Evaluation in the Real World: A comparison of Four Techniques". In Proc. of ACM CHI'91, ACM Press, 1991.

[Nie93] Nielsen J. "Usability Engineering", Academic Press, New York, 1993.

[Nie94] Nielsen J., Mack R.L. "Usability Inspection Methods", John Wiley & Sons, New York, 1994.

[NGA98] National Gallery of Art – Washington D.C., Web Site. Http://www.nga.gov. 1997.

[Reu94] Reunion des Musées Nationaux, Montparnasse Multimedia. "Le Louvre" CD-ROM, 1994.

[Reu96] Reunion des Musées Nationaux, Montparnasse Multimedia. "Musèe d'Orsay" CD-ROM, 1996.

[Scw97] Schwabe D., Rossi G., Garrido A. "Design Reuse in Hypermedia Application Development". Proc. of Hypertext '97, Southampton, UK, April 1997.

[UIE97] User Inerface Engineering Group. "Surprise on the Web: Results from Usability Testing". Http://world.std.com/~uieweb/surprise.htm,1997.

[Vat98] Musei Vaticani - Rome, Web Site. Http://www.christusrex.org/www1/vaticano/0-Musei.html. 1998.

[Yam95] Yamada S., Hong J.K., Sugita S. "Development and Evaluation of Hypermedia for Museum Education: Validation of Metrics". *ACM Trans. on CHI*, 2 (4), December 95.