Hypermeca Design Meth couldogy in World Wide Web Applications

Antonio Moreno-Muñoz
Adolfo Plaza-Alonso
Departamento de Electrotecnia y Electrónica, Universidad de Córdoba
Córdoba, Spain

Carlos de-Castro-Lozano
Departamento de Matemáticas Aplicadas, Universidad de Córdoba
Córdoba, Spain

Sebastián Dormido-Bencomo
Departamento de Informática y Automática, Facultad de Ciencias
Madrid, Spain

One approach to instruction based on a "learner-centered" view of learning is to provide rich environments in which learners can actually build their own knowledge. Therefore only educational software, carefully designed, can improve the efficiency of courseware. Within this framework, the World Wide Web provides a unique support for course material. Using a generalized hypermedia instructional design methodology, we integrated a hypermedia instructional module with simulation additional tools into an academic information system. This academic information system is a computer-supported environment in which collaborative discourse is the primary medium for knowledge advancement in the area of power electronics. It can be explored by learners in that they have both contextual access to knowledge displayed in a hypertext format and access to real experiences by means of simulation. In this article, we present an object-orientated approach that integrates the complete graphic user interface development.

1. INTRODUCTION

Nowadays, our students receive (and have to store) overwhelming amounts of information from the Internet. Therefore it is important to develop information-rich learning environments in which learners can select resources from which to learn.

Requests for reprints should be sent to Antonio Moreno-Muñoz, Department de Electrotecnia y Electrónica, Universidad de Córdoba, Avenida Menendez Pidal sn, 14008 Córdoba, Spain. E-mail: el1momua@uco.es
Using hypermedia capabilities offered by the World Wide Web (WWW), we may provide students with information in a wide variety of forms and enable them to seek their own path through the information channels. It is in this context that we decided to analyze the potential of the WWW as an innovative provider of academic information (Moreno-Muñoz et al., 1994). Although a tremendous amount of knowledge about hypermedia and hypertext systems has been accumulated (Bieber, Vitali, Ashman, Balasubramanian, & Oinas-Kukkonen, 1997; Nielsen, 1995), there is still only a small number of publications specifically dedicated to the Web as a teaching tool (Benyon, Stone, & Woodroffe, 1997). This article reflects our experience in using a particular design methodology, the so-called Object-Action Methodology presented elsewhere (Moreno-Muñoz et al., 1997a). This method can be applied not only to Internet-based educational environments but also to all hypermedia applications in general. To illustrate the hypermedia instructional design methodology, we use an academic information system that has been described elsewhere (Moreno-Muñoz et al., 1997d). Through the academic information system, students have access to general departmental information, displayed in a hypertext format. They also have preferential access to course materials. The subject selected here is the “Power Electronics Converters” course, from the industrial electronics degree at the University of Córdoba. Simulation tools have been integrated jointly with a hypermedia-based instructional module in a computer-supported environment in such a way that collaborative discourse becomes the primary medium for knowledge advancement in the area of power converters.

In this article, we confine ourselves to analyzing the way the information related to the department and most particularly the course materials for students are structured and organized. The outline of this article is as follows. The hypermedia design methodology employed is presented in Section 2. Section 2.1 is devoted to the analysis phase. Section 2.2 describes the data model conceptual design. Sections 2.3 and 2.4 describe the functional design and formal interface design. In Section 2.5, we present the WWW site prototype. In Section 2.6 the evaluation analysis are discussed. Finally, in Section 3, we present the conclusions.

2. HYPERMEDIA DESIGN METHODOLOGY

We can describe the “interaction component” or “interactions aspects” of an interface as the various ways in which a user interface works, its behavior in response to what the user sees, hears, and does while interacting with computers. The “interface software” includes the means for implementing the code that puts in motion the interaction component. To emphasize this distinction, the terms behavioral domain and constructional domain have been coined (Hartson et al, 1989). In the behavioral domain, interaction is described abstractly, independently of software, in terms of the users’ behavior and the interface as they interact with each other. In conventional development, software engineers and programmers have often designed the interaction components. The result has been a wide variety of interfaces with different levels of quality and usability. At this point it is necessary to select design features that not only bring quality appearance, but also account for the following (Marcus, 1995; Moreno-Muñoz et al., 1996, 1997b, 1997c):
- A comprehensible metaphor.
- Appropriate organization of data, functions, and tasks.
- Efficient navigation between data, functions, and tasks.
- Effective interaction sequencing.

Hypermedia networked courseware differs typically from traditional projects in several critical dimensions. First of all, these projects may involve people with different sets of skills: content writers, pedagogical and communication experts, graphics artists and designers, as well as programmers. On the other hand, a hypermedia project implies developing the structure of a complex domain, making it clear and accessible to users. Furthermore, the rich variety of multimedia data types in hypermedia applications raises numerous difficulties (Hardman et al., 1994). Finally, the need for prototyping and intensive testing with users is even more pronounced in hypermedia development than it is in traditional applications, because user tolerance to errors is very low here (Isakowitz, Stohr, & Balasubramanian, 1996).

From the point of view of the relation between cognition and hypermedia, it is necessary to distinguish between two kinds of applications. First, applications that can be freely explored by readers appear as browseable databases or “hyperbases.” In contrast, applications that are directly tied to specific problem-solving situations appear as complex electronic documents or “hyperdocuments” and are quite structured and even constrained (Wright, 1991). Whereas the first are best suited to support unconstrained searching and information retrieval, the second are better suited for tasks requiring deeper understanding and learning.

We decided to use an object-oriented design paradigm, because its abstraction and composition mechanism presents a concise description of complex information items and, in addition, provides a particular emphasis on navigational and formal interface design. The techniques developed here have to be thought as describing interactions from the user’s viewpoint. This methodology leads to the application of the “star life cycle” user interaction model (Harton et al, 1989) in hypermedia application development (see Figure 1). This life cycle presents an

![Figure 1: Star life cycle.](image-url)
evaluation-centered approach; the various activities are interconnected through the evaluation tasks in the center. The designer can start at any point in the cycle, evaluate the process, and continue development at any other branch. In Table 1 we summarize the Object-Action Methodology activities and modeling approach. In this section we focus only on an “informal description” of the design phases method.

2.1 Analysis

The first phase in the process is the application analysis. It includes typically: needs analysis, user analysis, task analysis, and data analysis.

**Needs analysis.** What does the target audience expect from the new application? The most direct way of finding out is to conduct a survey or at least to ask some representative members of the group.

The goal of our application is to provide information related to programs, courses, and internal and external activities.

- Assumption: The context is the University of Córdoba. Even if the Internet were not used as the primary exposition of material, we can assume that tradi-

<table>
<thead>
<tr>
<th>Phase</th>
<th>Products</th>
<th>Process</th>
<th>Method Concern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>Specifications sheets</td>
<td>Requirements gathering</td>
<td>Analysis of needs, potential users, data, and task</td>
</tr>
<tr>
<td>Data model conceptual design</td>
<td>Classes, subsystems, relation, attributes</td>
<td>Aggregation, generalization, and specialization</td>
<td>Semantics of the problem domain</td>
</tr>
<tr>
<td>Functional design</td>
<td>Nodes, links, and access structures</td>
<td>Mapping between data model and navigational objects</td>
<td>Metric of design</td>
</tr>
<tr>
<td>Physical design of interface</td>
<td>Formal interface objects, responses to external event, interface transformation</td>
<td>Mapping between navigational and perceptible objects</td>
<td>Modeling perceptible objects, implementing chosen metaphors</td>
</tr>
<tr>
<td>Rapid prototyping</td>
<td>Horizontal and vertical prototype</td>
<td>Compromise between prototyping and evaluating</td>
<td>Allows requirements to be demonstrated in the prototype</td>
</tr>
<tr>
<td>Deployment</td>
<td>High fidelity running application</td>
<td>Multimedia deployment, integration, and implementation</td>
<td>Complete functionality</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Usability documents</td>
<td>Predictive and formative evaluation</td>
<td>Quantitative and qualitative data generation</td>
</tr>
</tbody>
</table>
tional lecture-based exposition could change to design-example classes or problem-based classes.

- Features: to integrate and structure the information needed by the student. Direct access to the various media employed in courses (such as text, animation, interactive simulation, and so on).

**User analysis.** Much as we may wish it would, the application will not appeal to everyone. Thus the best option is to choose a particular segment of the population and focus efforts on these users. One could begin exploring these questions: Who is likely to be interested in the content the application will provide? Which of these user groups is the application equipped to serve? Of the remaining list, which potential users are most likely to have access to the Web? The target audience definition will influence every aspect of the application design. Based on potential user interviews, it becomes feasible to gather information about the characteristics and necessities of possible users. Some of the data gathered can be presented as follows:

- General education and experience: appropriate levels of complexity and formality.
- Expertise level (novice, intermittent, or frequent user).
- Input and output media: voice, visual, or both.
- Skill base (e.g., typing).

Returning to our application, there are several things that could be said about expected users:

- User characteristics: university students with experience and skills in general computer software for engineering (e.g., CAD, simulation packages, spreadsheets, or databases). Medium or high level of formality and complexity in language expression is required. Professional and technical approach is needed.
- Skills: High general skill level. Some keyboard or typing skill is beneficial but not positively required. Simulation program understanding is needed.
- General advises: Keep it austere, simple, and transparent. Establish easy-to-follow structure.

Task analysis specifies the functionality of the application, indicating the task and function allocation. As in Hartson et al. (1989), we make use of task analysis to assist in drive design. A hierarchical set of tasks has to be identified; the way users have to utilize the application is then established. It is often useful to start with a single high-level task and decompose it into the next level down. At this point, developers are often compelled to make decisions about the design for the first time. This interdependency between task analysis and design is the main reason for transferring the task study to the design phase as shown in the following.

Finally the data analysis allows us to specify the data involved and the relations among them. If we consider the university department as an information organization in which instructors and students collaborate actively in the learning process,
our academic information system will be the support for "human–human interaction" bridging distance and time.

We can now determine the data structure required:

- Workspace: provides the physical support to this information-distributed structure.
- External activities: R&D programs, collaboration with companies and institutions, intensive seminars, links to other information resources, and so on.
- Publications: books and papers published by the community.
- Faculty: information about the department staff.
- Projects: works developed by the community.
- Syllabus: in which the department collaborates.
- Courses: in which the department participates.

The full application demonstrates the natural hierarchy of the course from "sections" to "section units," which allows us to identify the major entities of the course: aims, objectives, topics, tutorial, and laboratory practices. The hypertext is made out of topics. The tutorial—a problem-based environment—is built on the Collaborator Notebook. This is a powerful tool integrated in a text processor that solves mathematics expressions through MATLAB®. The Collaborator Notebook assists students in analyzing the quantitative aspects of power converters such as harmonic analysis, discontinuous conduction, rectify-inverter mode, or improvement of the efficiency factors. While solving problems, students have access to Simulink® simulator. Simulation helps expert users understand the converters' performance, and in addition allows novices to integrate knowledge and concepts through the guided project study. Laboratory practice permits students to practice through computer instrumentation tools and compare these with the results of the tutorial (Moreno-Muñoz et al., 1997d).

### 2.2 Data Model Conceptual Design

The approach presented here differs from others used in hypermedia systems, which detail their internal architecture but are of little value in modelling hypermedia applications (Isakowitz et al., 1995). In this phase, an object-oriented data model is described using a well-known object-oriented methodology (Coleman et al., 1994). This modelling approach is improved by means of emphasizing object structure rather than object behavior. Its purpose is to capture the concept that exists in the domain of the problem. Problem domain design falls into the behavioral domain, in that it is independent of software design and implementation (Hartson et al, 1989). The model is built on classes, attributes, relations, and subsystems. The extensions allow the use of aggregation, generalization, and specialization. Figure 2 shows the schema of the complete academic information system. Boxes represent classes with name and attributes, arcs represent relations, and the small diamond at the end of an arc indicates aggregation.
In this application we have defined four main classes—Department, Publications, Projects, and External Activities—and three subsystems—Courses, Professors Staff, and Workspace. Classes inherit attributes, parts-structure, relations, and behavior of their super-classes. In Figure 2, the relation between Course and Programs reflects the fact that instances of each subclass, Topics (with its subtopics) or Tutorial, belong to a Syllabus.

2.3 Functional Design

The purpose of the functional model is to describe the behavior of the application. Functional design is expressed by defining two schemes: the navigational class schema and the navigational model. The navigable objects or “hyper-components” classes are nodes, links, access structures, and so on. Node class is derived from data model classes, and links are the navigational realization of relations defined in the same schema. The semantics of nodes and links are the usual ones in hypermedia applications and other access structures, such as “indexes”
and "guided tours," which represent possible ways of accessing nodes. As usual, the guided tours are navigated through "next" and "previous" buttons for those wishing to follow a sequential ordering.

Different functional models can be built from the same data schema, thus expressing different views of the same information base. For example, a node class does not need to contain all the attributes of the data model class, as they represent information of no interest to the audience. Furthermore, attributes with multiple perspectives become different attributes of the node class; for instance, in the Department node "Location.string" becomes "Location" and "Location.figure" becomes "Image." It should also be noted that node classes include "Anchor" attributes for all the outgoing links, which in turn are views of the relations in the data model.

### 2.4. Formal Interface Design

Once the functional structure has been defined, it has to be perceptible to the user through the application interface model. This phase is built around the User Action Notation (UAN) technique (Hix & Hartson, 1993). The UAN is effective, precise, and concise in describing user tasks. Other task-oriented models exist, but they are originally oriented toward analysis. In contrast, the UAN directly supports design.

![FIGURE 3 Navigational classes.](image-url)
We decided to complete UAN task description in two steps: the configuration scenario and the task-oriented chart.

The use of configuration scenarios allows us the specification of the interface objects, their structural relation, and the communication among interface objects and navigational objects (their owners). Particularly in hypermedia interaction, each node class will react to the “anchor selected” mandate by asking the anchor to follow the navigation through the link. For each navigational class, it is necessary to define its corresponding configuration scenario. In Figure 5, the “Topic public interface” can react to external action (mouse clicked, display or mouse double-clicked) and communicate with its owner (the node Topic) by sending the messages “anchor selected,” “get name,” and “get schema.”

The task-oriented charts are in some ways similar to the Operation Model Schemata (Coleman et al., 1994). Although task-oriented chart notation is behavioral-representation oriented, by using UAN terminology, this compacted expression of composite objects adds the behavioral and structural dimension to the schema. In Figure 6, “Topic public interface” may be either in state display “off” or display “on.” With the notations of the figure, state display “on” is tied to an “And” of 10 substates, corresponding to the node schema field, the text field, and buttons. Sometimes it is necessary to represent exclusive states that will not usually be perceived at the same time by the user. On these occasions we may use the “Xor” composition. As shown in the figure, transitions are annotated with the
task that causes the transition. The fundamental task "manage topic" is defined in terms of subtasks. The necessary user action and the interface feedback accompany each one. Although with UAN we can also specify more complex behavior, in this article we focus only on the behavior that activates navigation. Thus all tasks described here instantiated a "select object" task, which is the only one defined at an articulatory level.

2.5 The Prototype

As mentioned previously, the development process exemplified here is evaluation centered. Through quick prototyping it becomes possible to evaluate the proposed design very early in the life cycle and in this manner to ensure that usability is built into the evolving interaction design. One purpose of the earliest user interface prototypes is to test the overall interaction metaphor that we are designing.

Figure 7 shows a typical page of our HTML version. The complete version can be explored at http://www.uco.es/dptos/electro/.

2.6 Evaluation

In evaluation it is possible to contemplate two different techniques: summative evaluation and formative evaluation. Although the formative evaluation tests the application in collaboration with the users, other widely used techniques are grouped as predictive evaluation, in which the aim is to predict the kind of prob-
TASK: Manage Topic (1)
(Access applications | access rectifier | access chopper
| access inverter | access semiconductors | access next
topic | access previous topic | access power electronics)+

TASK: Access Applications (2)
USER ACTION: Select (Applications).
INTERFACE FEEDBACK: view level = display off

TASK: Access Rectifier (3)
USER ACTION: Select (Rectifier).
INTERFACE FEEDBACK: view level = display off

TASK: Access Chopper (4)
USER ACTION: Select (Chopper).
INTERFACE FEEDBACK: view level = display off

TASK: Access Inverter (5)
USER ACTION: Select (Inverter).
INTERFACE FEEDBACK: view level = display off

TASK: Access Semiconductors (6)
USER ACTION: Select (Semiconductors).
INTERFACE FEEDBACK: view level = display off

TASK: Access Next Topic (7)
USER ACTION: Select (Next).
INTERFACE FEEDBACK: view level = display off

TASK: Access Previous Topic (8)
USER ACTION: Select (Previous).
INTERFACE FEEDBACK: view level = display off

TASK: Access Power Electronics (9)
USER ACTION: Select (Power Electronics).
INTERFACE FEEDBACK: view level = display off

FIGURE 6 Task oriented chart.
lems that users will encounter when using the application without actually testing it with the users. Different methods are convenient for different stages of application development. Predictive evaluation methods can be applied very early in design. However, formative tests with users are usually done later on a global prototype. Several studies have compared different evaluation methods with the aim of identifying the most cost-effective and reliable ones (Nielsen, 1993). We decided to combine a predictive technique such as heuristic evaluation in an early prototype with a user attitude questionnaire in the last prototype, as a way of canvassing the subjective opinions of students about the software. This multifaceted approach ensures that a broad overview is obtained during evaluation and provides a suitable basis for recommendations about future work.

**Heuristic evaluation.** Jakob Nielsen (1993) defined heuristic evaluation as a usability inspection method for finding usability problems in a user interface. The procedure demands having a small set of evaluators to examine the interface and judge its compliance with recognized usability principles (the heuristics; Nielsen, 1993). The objective was to detect the usability problems of the initial prototype version and avoid them in the final prototype.

Two experimenters were in charge of managing the evaluation session. As a previous step, the experimenters established a list of usability principles or heuristics as shown in Table 2. The heuristics were derived from a general-purpose list given by Nielsen (1993), supplemented by cognitive-related hypermedia specific princi-
Table 2. List of Usability Principles

1. Use simple and natural language. User interface should be simplified as much as possible. The ideal is to present exactly the information the user needs, how, when, and where it is needed.
2. Speak the users' language. The terminology in user interface should be in the users' language and not in system-oriented terms.
3. Minimize the users' memory load. Do not force users to remember key information across the application. The computer should take advantage of recognition rather than recall.
4. Consistency. Similar information should always be presented through identical terminology. Follow user interface standard and principles of graphical communication.
5. Feedback. The system should continuously inform the user about what it is doing and how it is interpreting the user's input.
6. Flexibility. The application should accommodate user sophistication (novice and expert users) and diverse user goals.
7. Coherence. Visualize the structure of the document. Use higher order information units, indicating equivalencies between them.
8. Facilitate navigation. Include cues for the visualization of structure, which show the user's current position, the way that led to this position, and navigational options for moving forward.

...
ions and suggestions in relation to the potential improvements of the academic information system.

Following the debriefing session, the experimenters established a composite list of violations. One day later, the evaluators were asked to assign a severity rating to each violation on a 5-point scale; this task lasted 1 hr.

**Heuristic evaluation results.** Usability problems were found in the user interface prototype subjected to the heuristic evaluation. A wide variety of problems were found, ranging from features of individual pages to general usability principles. As shown in Table 3, severity ratings per heuristic were also varied. The lowest severity rating and a low variance were found in heuristic 1 (to use simple and natural language). Principle 6 shows a low severity rating, perhaps because it is not an actual problem here. The principles that show the highest severity rating are heuristics 2 and 8.

Although the application was found to be easy to use, the team found some lack of consistency in the look or style across the pages. The loading of some pages was too slow, requiring some kind of feedback message to keep the user informed. Regarding the violation of the second principle, even if the presentation of the pages was generally effective, the target audience study had not been completed properly so the contents should be improved in a more stylized and austere manner. There appears to be some lack of coherence, because evaluators found that the structure was not correctly prepared and there was not a complete definition of higher order information units. Furthermore, the evaluators’ familiarity with the contents allowed them to find some granularity-related problems. The utilization of some validation metric was recommended.

In addition, one major problem was found in a part of the interface that was not subjected to intensive evaluation. HTML on its own could not support the features that we required to provide an inexpensive engineering simulation tool for our students. Thus, we decided to use the simulation tools based on the well-known simulation tool MATLAB, although this option results in a lack of consistency in the interface. However, with Web solutions continuously improving and taking advantage of Java alternatives, this problem should soon find a solution.

<table>
<thead>
<tr>
<th>Principle</th>
<th>Total Violations</th>
<th>Avg. Severity</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use simple and natural dialogue</td>
<td>4</td>
<td>2.3</td>
<td>0.77</td>
</tr>
<tr>
<td>2. Speak the users’ language</td>
<td>17</td>
<td>3.19</td>
<td>1.06</td>
</tr>
<tr>
<td>3. Minimize the users’ memory load</td>
<td>3</td>
<td>3.00</td>
<td>0.79</td>
</tr>
<tr>
<td>4. Consistency</td>
<td>19</td>
<td>3.08</td>
<td>0.99</td>
</tr>
<tr>
<td>5. Feedback</td>
<td>15</td>
<td>3.77</td>
<td>1.87</td>
</tr>
<tr>
<td>6. Flexibility</td>
<td>21</td>
<td>2.6</td>
<td>0.56</td>
</tr>
<tr>
<td>7. Coherence</td>
<td>15</td>
<td>2.8</td>
<td>0.20</td>
</tr>
<tr>
<td>8. Facilitate navigation</td>
<td>22</td>
<td>3.57</td>
<td>0.46</td>
</tr>
</tbody>
</table>
Table 4. Results of the Attitude Questionnaire

<table>
<thead>
<tr>
<th>Question</th>
<th>Avg. Severity</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Characters on screen</td>
<td>8.5</td>
<td>0.54</td>
</tr>
<tr>
<td>2. Image of characters</td>
<td>8.5</td>
<td>1.04</td>
</tr>
<tr>
<td>3. Character shape (fonts)</td>
<td>8</td>
<td>1.09</td>
</tr>
<tr>
<td>4. Was highlighting on the screen helpful?</td>
<td>7.5</td>
<td>1.51</td>
</tr>
<tr>
<td>5. Were screen layouts helpful?</td>
<td>7.83</td>
<td>1.47</td>
</tr>
<tr>
<td>6. Amount of information that can be displayed</td>
<td>7.66</td>
<td>0.51</td>
</tr>
<tr>
<td>7. Arrangement of information on screen</td>
<td>8.16</td>
<td>0.98</td>
</tr>
<tr>
<td>8. Sequence of screens</td>
<td>8</td>
<td>1.41</td>
</tr>
<tr>
<td>9. Next screen in sequence</td>
<td>8</td>
<td>1.09</td>
</tr>
<tr>
<td>10. Going back to previous screen</td>
<td>7.66</td>
<td>1.21</td>
</tr>
<tr>
<td>11. Definition of tasks</td>
<td>8</td>
<td>1.41</td>
</tr>
<tr>
<td>12. Were the buttons visible?</td>
<td>6.33</td>
<td>2.33</td>
</tr>
<tr>
<td>13. Were the names of the titles adequate?</td>
<td>7.66</td>
<td>1.36</td>
</tr>
<tr>
<td>14. Comprehension of menus</td>
<td>8</td>
<td>1.67</td>
</tr>
<tr>
<td>15. Flux of interaction</td>
<td>8.33</td>
<td>0.51</td>
</tr>
<tr>
<td>16. Required performance</td>
<td>7.83</td>
<td>1.32</td>
</tr>
<tr>
<td>17. Response of the system</td>
<td>7.83</td>
<td>1.16</td>
</tr>
<tr>
<td>18. How were the results presented?</td>
<td>6</td>
<td>3.34</td>
</tr>
<tr>
<td>19. Presentation of contents</td>
<td>8.16</td>
<td>0.98</td>
</tr>
<tr>
<td>20. How were the options presented?</td>
<td>7.83</td>
<td>0.98</td>
</tr>
<tr>
<td>21. Formality in terminology</td>
<td>8.66</td>
<td>0.81</td>
</tr>
<tr>
<td>22. Organization of information</td>
<td>8</td>
<td>0.63</td>
</tr>
<tr>
<td>23. Progress in level of information</td>
<td>8</td>
<td>1.26</td>
</tr>
<tr>
<td>24. Contents</td>
<td>6.83</td>
<td>3.43</td>
</tr>
<tr>
<td>25. Did you feel the system under control?</td>
<td>7.66</td>
<td>1.50</td>
</tr>
<tr>
<td>26. How interesting did you find using the system?</td>
<td>8.5</td>
<td>1.04</td>
</tr>
<tr>
<td>27. How beneficial did you find using the system?</td>
<td>8.5</td>
<td>0.83</td>
</tr>
<tr>
<td>28. How varied was the system?</td>
<td>7.66</td>
<td>0.81</td>
</tr>
<tr>
<td>29. Were you tired after using the system?</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>30. Were you passive or active using the system?</td>
<td>8.5</td>
<td>1.04</td>
</tr>
<tr>
<td>31. How useful did you find this kind of system?</td>
<td>8.16</td>
<td>1.16</td>
</tr>
</tbody>
</table>

**Attitude question evaluation.** Forty students took part in the trial. They were instructed to explore the prototype without specific limitations. Students were asked to complete a questionnaire aimed at promoting their views and opinions about the package in terms of its effectiveness, the level of use interest, design aspects of the presentation, ease of use, and overall subjective rating. See the appendix for the complete questionnaire. The results obtained are reported in Table 4.

Student attitude evaluation allows us to draw interesting conclusions with regard to the subjective acceptance of the application and may also present aspects of the system stiffly identified by the previous evaluation test. It is important to assess subjective acceptance, because an application that is not well received by its users is unlikely to be successful in the long run, however good it may appear when judged by other criteria (Waterworth, 1989). The overall ratings were frankly positive. This suggests that the application was well improved from the first prototype initially evaluated by the method of heuristic evaluation.
The low ratings detected in questions 18 and 24 are related mainly to the presentation of results and contents and reflect perhaps the students' low level of familiarity with the displayed material. In question 12, some of the students found the "hot" or "active" zones hard to see. This is likely to be due to the fact that many graphic user interfaces overuse shadows, animations, and so on in buttons or active zones. In our opinion, this tendency is becoming outdated and superseded.

Particularly positive ratings were elicited by questions 1, 2, 26, and 27. The high rating detected in question 21 shows that one of the objectives stated has been reached. We can conclude that the users judged the application to be reasonably well designed, particularly in terms of understandability, systematic arrangement, and suitability of purpose. Several students offered useful comments on how to improve the effectiveness of the application.

3. CONCLUSIONS

In this article, we have used a design methodology, the so-called Object-Action Methodology, and discussed each of its basic steps. With this methodology, it is possible to build a schema for a domain of hypermedia applications. Using functional classes, it is possible to provide a smooth transition from domain and application modeling to concrete hypermedia design. Working with the formal interface model, it is feasible to map the hypermedia objects defined as navigational classes into perceptible objects.

Another author has also proposed a formal model, the Objects/Actions Interface Model (Shneiderman, 1997). This model encourages designers to focus on four components: objects, actions, metaphor, and handles for actions. Our work is different because it clearly separates navigational behavior from interface specification. Moreover, task-oriented charts not only indicate composition by behavior, as in state chart, but also structural composition.

Our evaluation tests show that this application is able to support successfully the information provided by the university department. However, the constraints imposed by the lack of integrated and adequate software engineering tools, both in terms of authoring and delivery, remain a major obstacle in developing WWW-based courses. Comparing the prototype tool with others previously used by the authors (for instance, Toolbook® or Microsoft Multimedia Viewer®), we found in the HTML-based tools a relative lack of control for the exact page size. It is possible to recognize more generic limitations in hypertext Web technology (Nielsen, 1996). We have solved the problems detected by other authors (Benyon et al., 1997) related to the level of formatting required in terms of colors and screen layout consistency by the use of the NetObject® fusion developers program. Unfortunately, the necessity of providing a stable frame for academic development prevented us from using evolving technologies.

In our opinion, as demand for hypermedia tools increases, there is a need to replace the current development approach, which is highly labor intensive and costly, with a more efficient approach that integrates the user requirements. The
purpose of this article was to present a hypermedia design methodology that addresses these issues.

REFERENCES


**APPENDIX: USER ATTITUDE QUESTIONNAIRE**

In this quick test, we are trying to find out what you (the user) think about the application. Thank you for your help in completing this questionnaire. Please return it when completed.

**Instruction**

Please answer the questions by selecting a number on the scale, to show what you think. For example, in the question asked:

Now answer the following questions by marking the scale to show how you felt about the application you have just used.

<table>
<thead>
<tr>
<th>Was the screen layout helpful?</th>
<th>Never</th>
<th>Always</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3</td>
<td>(4) 5 6 7</td>
</tr>
</tbody>
</table>

**Remember: we are testing the application, not you.**

Try and avoid picking the No answer N/A part of the scale. Please answer quickly without spending too long on each question. Answer all the questions.

**Personal questions**

Which course(s) are you studying this year?
Which course(s) did you study last year?
What sort of software do you usually work with?
What kind of computer do you usually work with?

**User Attitude Questionnaire**

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Hard to read</th>
<th>Easy to read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characters on screen</td>
<td>1 2 3 4 5 6 7</td>
<td>8 9 10 N/A</td>
</tr>
<tr>
<td>Image of characters</td>
<td>Fuzzy</td>
<td>Sharp</td>
</tr>
<tr>
<td>Character shape (fonts)</td>
<td>Barely legible</td>
<td>Very legible</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
<td>8 9 10 N/A</td>
</tr>
</tbody>
</table>
Was highlighting on the screen helpful? | Not at all | 1 2 3 4 5 6 7 | Very much | 8 9 10 N/A
---|---|---|---|
Were screen layouts helpful? | Never | 8 9 10 N/A | Always | 8 9 10 N/A
Amount of information that can be displayed. | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Arrangement of information on screen | Illogical | 8 9 10 N/A | Logical | 8 9 10 N/A
Sequence of screens | Confusing | 8 9 10 N/A | Predictable | 8 9 10 N/A
Next screen in sequence | Unpredictable | 8 9 10 N/A | Predictable | 8 9 10 N/A
Going back to previous screen | Impossible | 8 9 10 N/A | Simple | 8 9 10 N/A
Definition of tasks | Confusing | 8 9 10 N/A | Clearly marked | 8 9 10 N/A
Were the buttons visible? | Not at all | 8 9 10 N/A | Very much | 8 9 10 N/A
Were the names of the titles adequate? | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Comprehension of menus | Bad | 8 9 10 N/A | Good | 8 9 10 N/A
Results of selections | Unexpected | 8 9 10 N/A | Expected | 8 9 10 N/A
 Flux of Interaction | Continue | 8 9 10 N/A | Discontinue | 8 9 10 N/A
Required performance | Very much | 8 9 10 N/A | Very few | 8 9 10 N/A
Response of the system | Slow | 8 9 10 N/A | Quick | 8 9 10 N/A
How were the results presented? | Confusing | 8 9 10 N/A | Clear | 8 9 10 N/A
Capacity of the system | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Presentation of contents | Confusing | 8 9 10 N/A | Clear | 8 9 10 N/A
How were the options presented? | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Formality in terminology | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Organization of information | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Progression in level of Information | Inadequate | 8 9 10 N/A | Adequate | 8 9 10 N/A
Contents | Incomplete | 8 9 10 N/A | Complete | 8 9 10 N/A
Did you feel the system under control? | Never | 8 9 10 N/A | Always | 8 9 10 N/A
Apprenticeship | Very Boring | 8 9 10 N/A | Very interesting | 8 9 10 N/A
How interesting did you find using the system? | Not at all | 8 9 10 N/A | Very much | 8 9 10 N/A
How beneficial did you find using the system
<table>
<thead>
<tr>
<th>Question</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>How varied?</td>
<td>Not at all</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td></td>
<td>Very much</td>
</tr>
<tr>
<td></td>
<td>8 9 10 N/A</td>
</tr>
<tr>
<td>Were you tired after using the system?</td>
<td>Very much</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td></td>
<td>Not at all</td>
</tr>
<tr>
<td></td>
<td>8 9 10 N/A</td>
</tr>
<tr>
<td>Were you passive or active using the system?</td>
<td>Passive</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td></td>
<td>Active</td>
</tr>
<tr>
<td></td>
<td>8 9 10 N/A</td>
</tr>
<tr>
<td>How useful did you find this kind of system?</td>
<td>Not at all</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td></td>
<td>Very much</td>
</tr>
<tr>
<td></td>
<td>8 9 10 N/A</td>
</tr>
</tbody>
</table>