Abstract

Teaching architecture is not primarily an instructional process but rather a process of interaction and experience. In this context multimedia material can be used to provide an active educational environment where students learn by doing. To yield an effective learning system expertise from various fields have to be combined. This paper emphasizes the technological challenges of multimedia applications in architectural design education. We discuss two research prototype systems and analyze the influence of the underpinning technology on the performance of the overall system design. Finally we give technical requirements that are demanded for next generation systems and propose a framework for concerted research action.

Introduction

Although multimedia is a commonly used term these days it is still difficult to define. People coming from diverse research disciplines will most likely link it to various meanings. We will stick to the following definition: multimedia-enabled applications in computer systems deliver and present continuous synchronized media (Rodriguez and Rowe, 1995), e.g. audio and video. This contrasts other definitions that associate a richer functionality to the term multimedia. For an active educational software environment (e.g., Schank, 1994), multimedia is only one building block in the puzzle of technologies that have to be integrated to yield an effective software system. Other techniques used originate from various fields including (but are not restricted to) computer-supported cooperative work (CSCW), software engineering, artificial intelligence, human-computer interface design, and pedagogic coupled with expertise in the domain under consideration. Maybe it is the resulting complexity that prevents teaching software from being open to reuse but rather being more or less constructed each time from scratch.
In the following we will discuss the role of technological issues in so-called multimedia applications for architectural design education. We will draw on experience that we gained throughout our work on a design-supporting environment for students of architecture. We will analyze two research prototypes of our system HySAT (Hypertext System for Architectural Typology) (Kühn and Herzog, 1991, Herzog, 1994) and conclude by postulating technical requirements that we think have to be fulfilled to yield an effective system.

Use of multimedia
Audiovisual material can serve a variety of goals within educational software in architecture. If we look at the didactic of design education, multimedia material is often used to represent aspects of architectural design artifacts that would otherwise be hard to communicate. Architectural design education is at its core about the design and analysis of buildings and their interactions with the surrounding environment. For this purpose, objects of the real world have to be represented and abstracted in, e.g., drawings, images, two- and three-dimensional models, along with explanatory text. For any architectural discourse, design artifacts (projected as well as realized) play an important role being used as "objective" basis to support subjective arguments in the discussion. Some examples of multimedia representation of real world objects can be found in, e.g. (Marinelli, A.M., Bilibani, R., and Gadola, A., 1994, or Sabater, T. and Gassull, A., 1994).

**Knowledge Level**

This use of multimedia material is an instance of the generic class of multimedia information systems. Information systems have a long tradition in decision making domains. Their purpose is to gather data and knowledge that in turn is used to justify certain actions carried out in the real world. We will follow the distinction between data and knowledge as given by Wiederhold (Wiederhold, 1992): "Data describes specific instances and events. The correctness of data can be checked versus the real world. Knowledge describes abstract classes. Experts are needed to gather and formalize knowledge." For multimedia information systems, data are the collection of plain media samples, whereas knowledge is made available by explicit representations of relationships among the content of those media samples. Figure 1 shows this conceptual distinction between data and knowledge.

In current implementations of multimedia applications in architecture, these boundaries between data and knowledge representations often get blurred. This hinders the successful reuse and sharing of already captured resources. Ng and Mori (Ng and Mori, 1994) come to a similar conclusion, where they call for a development of a mechanism whereby efforts can be pooled and shared. In our understanding such a
mechanism must be based on a model of information processing, that is open to different strategies of knowledge acquisition and reuse of multimedia resources. In the following we will discuss pitfalls and shortcomings that we experienced throughout the development of our prototype system HySAT that should be avoided in next generation systems.

The conceptual model of HySAT

Before we actually begin to discuss technical issues we will shortly review the conceptual model of our system. The main emphasis of our considerations lies on building a learning environment where students learn by doing. This idea is tightly coupled to the curriculum of architecture where first-hand experience is gathered in design studios. In contrast to rather passive systems that basically mimic "school books", our system aims at engaging the user in an active construction and transformation process of the system’s content. For a more in depth discussion of our didactic conception see (Kiihn and Herzog, 1994).

The main purpose of our project is that of delivering a design-supporting tool. This tool will help students to define and discover criteria that are important for a given design problem. Having understood the relationships, constraints, dependencies, and properties within the space of possible solutions, a corresponding solution pattern – a design solution - will be found more easily. For an interaction metaphor with the system we chose the exploration process that has been formally described by Smithers and Troxell (Smithers and Troxell, 1990). A more detailed discussion on the topic of exploration in multimedia document bases can be found in (Herzog, Petta, and Kuhn, 1995).

The current implementation of our system uses multimedia components and hypermedia documents to represent design artifacts. These documents are the core material of our information base. Students will extend the content of the system by adding new material in the course of interaction. The exploration process draws on an access mechanism that is used to abstract the content of these documents. Referring to our model of information systems given above, the multimedia components are part of the data layer, whereas the hypermedia and access mechanism components are settled in the knowledge level. This distinction is important, as others might use the same multimedia components for different purposes (e.g., courseware design) by using different technologies on the knowledge level.

Implementing HySAT using HyperCard

Our first prototypical implementation used HyperCard as programming environment. HyperCard has proved useful for rapid prototyping that allows for a first usability evaluation of our conceptual model. The HyperCard application has been used with students to gain feedback from prospective users. Figure 2 shows a screenshot of the HySAT environment. This version of HySAT is composed of three classes of stacks (in HyperCard, a stack is a collection of hypermedia nodes together with scripts to facilitate human-computer interaction): the text, the picture, and the plan stack, which are designed for presentation of corresponding media types (see Figure 3 for a functional diagram). Additionally, the linker stack implements a linking mechanism that allows users to link arbitrary chunks of information (a feature that is not readily available in HyperCard).
During the design course, students chose buildings on which they had to give a hypermedia presentation. This material also included an analytical design critic part. Our intention was to later reuse the material for the design-supporting tool. Students used the editing functions implemented in the HyperCard application to create their documents. The experience we gained during the course proved the assumption that generating hypermedia documents is more complicated - both, from a craftsman’s (e.g., how to prepare the diverse media components) as well as from an intellectual (e.g., what kind of narration and link structure is best suited) point of view - than the average student is able to handle without dedicated training.

Although we used HyperCard to implement just a part of the conceptual model as presented above, we soon came across technological barriers that we could not break. One of the main drawbacks of HyperCard is that it does not provide means for a sound distinction between the data and knowledge level of the system. Instead, a HyperCard application subsumes all aspects of user-interface design, functionality, and appearance of multimedia material in a proprietary format. Even worse, the technology does not allow to separate data from programs. For our approach of delivering stack templates this means that already existing stacks cannot be updated to new versions. Data are not stored transparent but rather hidden without direct control for the developer.

Another technological shortcoming of HyperCard is the lack for collaboration support between different users. HyperCard stacks cannot be used concurrently by users on a network. This fact does not correspond to our understanding of an active educational environment shared by all participants. Our work with students has shown that active support of collaboration is very important for the acceptance by users.
To conclude, this first prototypical implementation of our model proved viable for gaining a first "hands-on" experience. We could evaluate some assumptions and refine our theoretical approach towards the issue. In fact, the model we presented in the previous section had been improved based on these new understandings of the topic. We also could show that the principal design and conceptual model of our proposed approach towards educational software in the domain of architecture has potential for further development. We also discovered the limitations of the underpinning technology we had used to implement the system. It was the goal of the second prototypical implementation to overcome at least some of these drawbacks.

Implementing HySAT using World-Wide Web (WWW) technologies

The restraint to a non network-aware software has been the major bottleneck of the HyperCard prototype. The rapid spread of WWW-based document systems led us to consider this technology for the next implementation. The WWW project has been originally designed to support collaborators working on a common task. It allows a workgroup to access documents from remote sites in a seamless way. WWW documents are basically hypermedia documents that integrate different multimedia resources (e.g., images, drawings, animations, videos, scripts, etc.). Documents are written in the rather simple document description language HTML (Hypertext markup language). To facilitate the exchange and the referencing of documents across the network, an address schema (URIs, URLs) and a client/server network protocol (HTTP) have been introduced. This combination of a set of standards together with simple to use client and server software has set free the biggest digital "information wave" ever.

Figure 4 is a screenshot of the HySAT-W3 environment. On first glance the system does not look very different from our first prototype. It is still a collection of hypermedia documents that you can point and click on. However the technology behind the scenes is quite different. As mentioned above, these documents can be served from a server connected to a network and can be viewed on any computer platform that supports a viewing software. The barrier of the closed computer universe got broken down. Unfortunately there is also an opposite side of the coin. By switching to the new technology, we lost some of the functionality we developed for the HyperCard version. Moreover we lost all documents we had previously prepared using the HyperCard version because we could not convert them to the new format. The problem of legacy documents is very prominent in the field of multimedia applications. This results from the diversity of file formats used. The WWW version has the advantage of storing media components independent of the hypermedia documents. This allows for reuse of these components even if the
multimedia documents are abandoned. Another disadvantage - at least at the current moment - is the missing of appropriate authoring tools. This might change with further developments in WWW technologies. Nevertheless the need for powerful tools proved critical for any development of educational software. At the bottom line, every design is just as good as the resulting software and its usability for students. Without a well developed user-interface, students cannot concentrate on their task at hand. Instead, they have to struggle to operate the system. This in turn is often the reason why students lose attention and interest to work with the system. Overall the second prototype has some distinct advantages over the previous one, but still lacks important features to meet our expectations. In the case of WWW, most of these features could be integrated by improving the software tools. Still some features are restricted by the constraints of the technology (e.g., WWW defines only uni-directional links). This again shows the importance of a thoughtful development and evaluation of a model before choosing technologies to actually implement a system.

Technological issues

The evaluation of our development effort brings up the following topics for further research:

- treatment of legacy documents
- the ability to reuse material for different educational applications
- support for sharing of applications; in contrast to reuse, sharing occurs on the knowledge level whereas reusing takes place on the data level of the system
- transition to collaborative environments and network-based authoring tools.

The problem of legacy documents is mostly a pragmatic one. It can be solved – at least for a certain time period - by sticking to a certain set of formats or by providing translators from one format to another. Still, progress does not get to a halt. New formats and methods usually yield advantages over previous ones. Furthermore, the complexity of tools and formats have to be considered. Whereas simple file formats are rather easily reused in a changing environment, complex systems like databases are rather cumbersome to maintain. On the other hand, databases provide a richer functionality that in turn can be used to provide active means for reusing.
This leads to the topic of reuse. Capturing and editing multimedia material is coupled with costs. To decrease overall costs, reusing material is a good strategy. To achieve this, the system has to provide means to locate relevant material and to integrate that material in different applications. This calls for a commonly agreed upon method to make material available for all prospective users. This topic is an active research field in computer science and achievements can be adopted to gain technological advantages. The same issue, but on a higher abstraction level, is tackled with sharing. This is were intelligent methods fit in to help communicate the knowledge stored within the system. A variety of different approaches can be developed and tested. The accumulation of multimedia material throughout the use of the system will create an ideal testbed for this purpose. Methods have to support both acquisition and retrieval of knowledge and have to cope with emergent structures that cannot be fixed a priori. Sharing calls for a very flexible representation mechanism that can adopt to changing needs of the knowledge engineers.

Finally, interacting with multimedia material involves a lot of different activities, such as creating, organizing, and sharing work and ideas. The tools handed out to students have to support them in all of these activities. A network-based application should not only support the distribution of material but also provide communication channels to allow the discussion of the material.

**Conclusion**

Having seen the diversity of research issues involved in designing and realizing an educational multimedia environment, the natural choice for a conclusion is to highlight the need for a collaborative development, preferable by European Institutions. For a successful development, it is called for multiple expertise and resources. We see this thought in line with previous calls (Ng and Mori, 1994, Mirabelli et al., 1994). In our understanding, the critical point is bringing together both content and technology experts.

For partners involved in such a project possible workpackages might look like the following:

- definition of models for both the information and the information processing. Within these models, the use of conceptual levels has to guarantee an open, extensible approach
- based on these models, specification of proper technical standards that comply to already existing standards in the field
- implementing tools that correspond to the specification of the conceptual models and to technical standards
- using these tools for content engineering, evaluation, and observation of student behavior

Our experience – and that of others, too - revealed that a single development team will hardly be able to accomplish these tasks. This is especially true for the actual use of the system. Only a widespread use and a user community exceeding a "critical mass" will justify the massive development costs. Furthermore, without enough "content material", the evaluation of certain aspects such as intelligent retrieval methods will just not be possible. We think that a commonly accepted framework will gain room for individual research interests while not losing the view for the bigger picture.

**References**


Smithers, T. and Troxell, W. 1990. "Design is intelligent behaviour, but what’s the formalism?" AI EDAM 4, no.2:89-98.