OPEN ACCESS
Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Review Article

Development of a Hypermedia Authoring Tool for Interactive Web-Based Learning Environments in Computer

Seyfali Mahini[1*]

[1]Islamic Azad University, Khoy Branch, Khoy, Iran

DOI: 10.36348/sjet.2021.v06i07.002 | **Received:** 02.06.2021 | **Accepted:** 06.07.2021 | **Published:** 14.07.2021

***Corresponding author:** Seyfali Mahini

Abstract

The development of a Web-based learning environment is time-consuming and complex. Hypermedia and interactive aspects are problem-specific and usually require a special authoring tool. This article describes how the development of hypermedia environments for computer graphics is preceded by the development of a hypermedia authoring tool. The implemented architecture is presented, which meets the requirements with regard to the inexpensive construction of a new environment, its consistent modification, its step-by-step internationalization, personalization and the integration of interactive virtual experiments and learning methods.

Keywords: Computer graphics, hypermedia**,** hypertextual**,** Image processing, authoring tool.

1. INTRODUCTION

A lecture on computer graphics often lacks possibilities for visualization, especially when it comes to teaching algorithms. Conversely, the references to theory are often lost. Iin the exercises Hyperrmedia learning environments are intended to counteract this fact by integrating hypertextual lecture texts, programming interfaces and exercises into one Integrate frames and link them to one another. The integration of virtual experiments, which enable an intuitive, playful way of learning parallel to the theoretical or technical approach, is also important for understanding graphical algorithms [1, 2].

2. Gols

The creation of such a learning environment is complex. It should be structured uniformly and clearly and have good navigation options. In addition, it must be easy to modify and expand. For major modifications, authoring tools are required that keep the complex hypermedia structure consistent between the documents. Extensions such as internationalization or personalization (adaptability) must be planned in good time. Modifications of a document should be carried out online by the respective author using server-side scripts browser and the type of network connection. Furthermore, with regard to offline use or a CD version, the documents are available statically and server-side dynamic document generation is only used for online modifications, complex inquiries or personalized data (Fig-1). For such goals, the development of a corresponding authoring tool is necessary. In this article the structure of such a tool is considered and the Creation of two hypermedia environments presented as case studies: the first takes on all the tasks of an electronic webmaster and the second provides a learning environment for the Image processing.

The only requirements are a standard browser and Java 2 on the server side with a JDBC / ODBC-compatible database; The client needs a Java2 plug-in for virtual experiments in image processing.

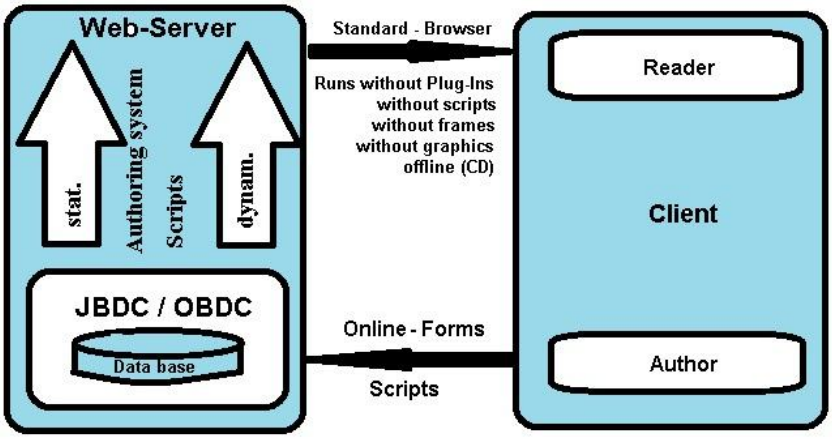

Fig. 1: Requirements for a web-based hypermedia environment

2.1 Terminology

The term document refers to media such as text, static images (digital, graphic), moving images (videos, animations), interactive virtual experiments (programs) and audio. A multimedia document consists of several such documents of different nature; if it contains references (links) to other documents, it is called hypermedial [3] and that resulting network referred to as the hypermedia environment. An interactive environment offers more complex and necessarily algorithmic capabilities that go beyond the navigation and search mechanisms of the display system (browser).

A teaching and learning environment also contains didactic exercises (fill in the blanks, multiple choice, free text, programming tasks, etc.) that consolidate the presented teaching content and provide feedback on the given solution. Automatic recording and correction is often not feasible in computer graphics and is compensated for by presence phases and manual processing. Ultimately, an authoring tool has the task of automatically generating such an environment from raw data and maintaining it in a consistent manner.

2.2 Classification

Typically, authoring tools provide a graphical software environment for developing hypermedia environments. The structure, content and design of such environments can be created visually with little effort and little prior knowledge using cut & paste or drag & drop. Script programs also allow more complex algorithmic processes. However, existing (commercial) authoring tools such as Asymetrix Toolbook or Macromedia Director do not cover the requirements mentioned [4, 5].

A conventional basis for hypertextual environments are open, platform-independent HTML documents that can be edited with any text editors or special HTML editors created, converted from other formats or dynamically generated by the web server [6].

HTML has neither semantic functionality like XML [4], nor does it offer synchronization and interaction means like SMIL [6]. This shortcoming is compensated for by Java applets and servlets. In this article, the former implement virtual experiments embedded in HTML documents, while the latter are similar to CGI scripts and implement server-side dynamic access to relational databases.

The use of the Java programming language for the development of the authoring tool and the virtual experiments is recommended because it is platform-independent, error-resistant and allows a object-oriented component model (Java beans) [7] with which exchangeable, reusable software components can be developed [8]. Though, they behave as a black box and allow structural learning exercises [2], but do not contain any standardized information about the semantics of the component [4]. A large number of standard tasks have already been implemented in Java packages, for example graphic interfaces in the Swing package [9] or functionality for image processing in JAI [6].

3. Building the architecture of a hypermedia authoring tool

When implementing the hypermedia authoring tool, structure, content and design were strictly separated. Each component is set up and modified in its own editor. The data itself, i.e. the structure table, text, references, multimedia documents and design notes are stored in a database. The final design of the documents and the navigation is determined by templates and the hypermedia environment is finally generated automatically by a generator component (Fig-2).

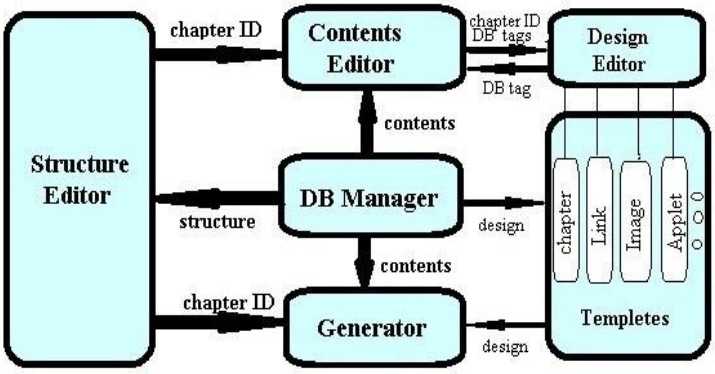

Fig. 2: Components of a hypermedia authoring tool

The hypermedia environment created will usually support several languages (internationalization). To achieve this, the languages used are layered on top of each other and the DB manager selects the appropriate layer depending on the active language; if the requested table does not exist, it changes to the next lower layer (Fig-3). In this way, the author can first create the content for one language and gradually add the content to other languages without difficulty. Language-independent data such as the structure table or images and videos without text or audio information are only kept in a single layer in this model.

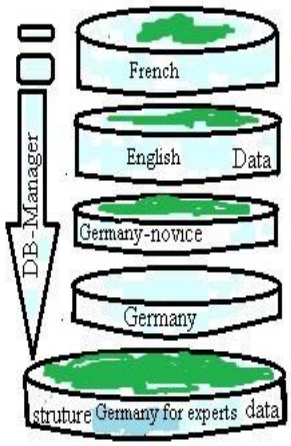

Fig. 3: DB layer model

Another application of the layer model results from the desired adaptation to the reader of detail. In Fig. 3, for example, the difficulty of the text is varied by layer for experts, standard users and novices.

3.1 Editors

The various editor components were quickly created with Java. A hierarchical directory structure component from Swing could be used for the structure editor. Also the construction and modification of the structure table from the database via cut & paste and drag & drop could be implemented with simple means.

Simple text components (ASCII) and WYSIWYG components (HTML) were also available for the text editor. The contents of the templates are compatible with the TML in the text Keyword addressed. For this purpose, the implementation of a design mode would be desirable in which these keywords would be displayed as template-dependent text components with appropriate functionality and thereby the text input also for authors which are not familiar with the syntax, would be faultless possible. In the current version of Swing, however, this task is very time-consuming. Finally, the design editor manages the registered templates and allows them to be modified the template parameter and represents the graphic surface of the active template.

3.2 Templates

Templates are used by the generator to produce the entire hyperstemed environment from the raw data and the structural table. Basic templates control the design of a single chapter, a reference, an image or a virtual experiment.

Templates for further multimedia content are conceivable and easy to integrate. With this separation of content and design, it is e.g. possible to fundamentally change the complete design of all screen pages by making small changes to the template. Images can be converted to other formats and with watermarks.

An algorithmic language is required to describe the documents to be embedded from the database, and two types of definition of templates have been implemented: templates can be programmed in Java or textually defined as a description similar to HTML. The latter implements a simple script language for database selection case distinction and document generation.

Depending on the type of data described, a template generates a graphic surface in which new data can be added or existing data can be modified. Also for these tasks were lists and tables components used by swing.

4. Case studies: Electronic Webmaster

The structuring of an institute homepage [10] using the authoring tool described went straightforward. The colors and pictograms of the templates for screen pages, links and images have been changed to the institute's own design. Then additional templates for people, work areas, projects, courses, offers, picture gallery and bibliography were designed. Sorted lists were described in more pure text, while in cases such as the person page, which lists all personal data in areas in which the person is active and does not generate a printable business card at the end. The Tmplate was programmed in Java.

The majority of the raw data could be imported into the database from other sources (annual reports, etc.) with the help of small transfer scripts. Only the area of bibliographies required a more complex script in order to also Import entries from a BibTeX file into the database.

After the generation of the hypermedia environment, a dynamic base servlet with authentication mechanism has been implemented to make online modifications from the author enable. In the event of a modification, the webmaster receives an email with the changes, which he is reviewing; if necessary, it starts the authoring system's generator immediately, otherwise the pages are generated automatically on a daily basis.

4.1 Learning environment for image processing

A learning environment for image processing is currently being set up, the first version of which was already described in [2].

When implementing the virtual experiments, it was often possible to reuse components from an earlier learning environment for computer graphics [KHS98]; the reader needs a browser plug-in for Java 2 for the virtual experiments. The actual task here is the teaching of graphic algorithms. Structural tasks in which the virtual experiment or subareas of them are based on basic components allow the knowledge acquisition through a data flow model.

4. SUMMARY

The structure of an authoring tool for the automatic generation of hypermedia environments was presented and attention was drawn to the inexpensive implementation in Java.

Structure, content and design were strictly separated A layer model of the underlying database enables step-by-step internationalization and the provision of various levels of detail. If possible, documents are generated statically and can be modified online. However, low-level programming tasks are solved in a programming environment separated from the learning environment and Make an evaluation difficult.

As next, templates can be designed for theoretical learning exercises such as gaps texts or multiple-choice issues These can be easily depicted as a synonym table or true-false table to the database.

REFERENCES

1. Owen, G. S. (1994). Teaching Computer Graphics as an Experimental Science. In Eurographics Workshop on Graphics and Visualization Education (GVE), Oslo, Norway, 10-11 September. Eurographics.
2. Hanisch, F., Klein, R., & Straßer, W. (1999). Ein Webbased Computer graphics course in a modular system, in: Martin Engelien and Jens Homann (eds), Virtual Organization and New Media - Workshop GeNeMe99, Eul-Verlag, Lohmar.
3. Halasz, F., & Schwartz, M. (1994). The Dexter Hypertext Reference Model. CACM, 37(2), 30-39.
4. Meissner, K., & Wehner, F. (1999). A document model for course documents in web-based virtual learning environments, in: Martin Engelien and Jens Homann (eds), Virtual Organization and New Media - Workshop GeNeMe99, Eul-Verlag, Lohmar.
5. SofTec NRW. (1999). Technical software requirements for multimedia teaching and learning systems, URL: http://www.uvm-nw.de, September 1999.
6. Sun. (1999). Java Advanced Imaging API White Paper (Beta), Sun Microsystems, Inc., April 1999 Mehr zu diesem AusgangstextFür weitere Übersetzungsinformationen ist ein Ausgangstext erforderlich Feedback geben Seitenleisten.
7. Graham, H. (1997). The JavaBeans API Specification. Sun Microsystems, Inc.,
8. Klein, R., Hanisch, F., & Straßer, W. Web based Teaching of Computer Graphics: Concepts and Realization of an Interactive Online Course, in: Michael Cohen (eds), SIGGRAPH 98 Conference Proceedings, Addison Wesley.
9. Eckstein, R., Loy, M., & Wood, D. (1998). Java Swing. O'Reilly & Associates, Inc.,
10. Institute for Interactive Graphics Systems (WSI / GRIS), homepage, URL: http://www.gris.unituebingen.de, University of Tübingen, 1999.