ACAM - an Active Context-based Authoring Model

Judith Kelner,
D. F. Hadj Sadok, Ricardo Martins.
Departamento de Informatica,
Universidade Federal de Pernambuco,
Cidade Universitaria - Recife - Pernambuco,
Caixa Postal - 7851,
CEP - 50732-970,
Brasil.

E-mail: {jk, jamel, rmas}@di.ufpe.br,
Tel: +55 (81) 271-8430,
Fax: +55 (81) 271-8038

Abstract

This work examines existing authoring models and methodologies. A new interactive, distributed, active, context-based authoring model is proposed.

The elaborated hypermedia model for distributed authoring is based on the concept of *contexts* and combines different authoring paradigms. Contexts define presentation policies and perspectives and offer a high level structural organisation of both the information and the authoring process.

Another contribution to research in the area made by this model, is its use of intelligent cooperating active agents to support run-time reference resolution, dynamic loading of applications and user interaction. The model adopts a modular multi-layered distributed architecture supported by the Sun's new Java object oriented language. The model makes clear division of the authoring process into two main stages, namely an authoring in the large and an authoring in the small phases.

Introduction

This work looks at the process of creating multimedia presentations, in other words, *multimedia authoring*. Continuing research is undertaken to develop new multimedia interactive models and many professional authoring products have already reached the market. Examples of these include Macromind's Director [Mac90], the Authorware package described in [NO94] and Harmony [ea91]. One major drawback of existing models and systems is their lack of high level adaptive use of the environment and distribution.

Hypermedia authoring involves the extra step of linking the authored information using a graph structure where links can be navigational, structural, semantical, etc. There is a wide spectrum of applications for authoring including: Interactive Multimedia Projects: describing systems, products, etc, Computer Based Training (CBT), Kiosks, Animations, Interactive TV and the Film industries, etc. Other applications include Training and Education also known as Teletraining, Simulation and Games, Sales and Advertising, Office Information Systems, Enginnering Documentation, Electronic Publishing, Medical Applications. More information on these applications may be found in [MG95].

Review

A more comprehensive review of existing Hypermedia and Multimedia models can be found in the authors' technical report[dASeKLS95]. This survey shows that existing hypermedia models are mostly generic and offer limited support to authoring and even those that do, such as HM-CARD[KAS94], often lack the necessary distribution and collaboration support and are usually far from being intuitive to learn how to use.

The discussion in [Pei95] includes HDM which is of particular interest to this work. This is an authoring model [GPS93] that allows the generic planning of hyper-documents. Its use of the concept of link perspectives, later described, is retained into the Active Context-based Authoring Model (ACAM).

Other *platforms*, such as those described in[ea93b] and [ea93a], support the use of re-usable and configurable active objects for the authoring of applications. While the authoring environment IMP (Interactive Multimedia Platform) suggested in [BB95] makes an attempt to address the entire spectrum of multimedia presentation it makes little or no effort for an analysis of the modelling of authoring systems.

Hypermedia based authoring systems include Harmony [ea91], Athens [VM93] and CMIF[1] [Bul91]. They support event driven functions used for building applications . These models together with other PC based ones such as Apple's Quicktime [Wan93] are mono-user systems and offer no

[1] The CMIF authoring model from CWI is based on the Harmony model.

	Dexter	Proxhy	Acqua	Hypercard	HDM	HM	OOHDM	Authorware
Concepts Simplicity	●	●	●	●	●	○		●
Semantics			○	○	○	○	●	
Application Modularity			○	○	○	●	●	○
Hypermedia Modeling	○	○	○	●	●	○	●	
Application Modeling			○	○	●	●	●	○
Cooperative Modeling			○					
BottomUp/TopDown					○	●	●	●
Security			○	○				
Version			●					
Design Reusability						●	●	

Table 1: Comparison of some Hypermedia Models: **none** not found ● found ○ insufficient

distribution support. Other standardisation efforts from MHEG (Multimedia and Hypermedia Expert Group) [MG95] and HyTime[ISO92][2] do not take into consideration the distribution aspects. This may change through the work of IMA [IMA93] nonetheless.

A brief comparison of some authoring models and systems is shown in the Table 1.

Requirements for Authoring Models

A more detailed list of hypermedia and multimedia application requirements can be found in the MHEG document [MG95]:

- system transparency to users, composition of multimedia data in time and space, synchronisation (elementary, chained, cyclic and conditional) and automatic adjustment;
- intuitiveness, simplicity of use and a short learning curve, object orientation and partial re-use of structures, information and projects;
- an authoring environment that makes it quick and easy to produce different levels of applications: simple presentations, hypermedia programs, and more sophisticated interactions with *answer judging*, feedback, and record keeping;
- *universal* access to retrieval services such as gopher, WWW, WAIS and multimedia database retrieval, support for international interchange standards such as MHEG and HyTime, real time interchange and presentation;
- a presentation service: with functions for sequential presentation, branching, conditional, query support, etc;

[2]HyTime is the official nickname given to the ISO/IEC international standard 100744: 1992, also known as the Hypermedia/Time-based Structuring Language”

- exploration, communication, media annotation and automatic generation of certain types of links, aggregation - an abstract mechanism used for referring to a collection of objects using a unique identifier;
- configuration, navigation and different interaction modes, quality of service (QoS) management and ability to build management policies;

Presentation and User Interaction Requirements

A number of the existing multimedia platforms, have little dealt with user interaction aspects. This is mainly due to the fact that only few of these are oriented towards authoring. Events are defined which represent asynchronous run-time operations as event functions. These include user interaction, system signals, etc. One of the novelties of the MHEG standard over existing multimedia interchange formats is its identification and encoding of user events as objects just like data objects including text, graphics, etc.

Context

The problem with using hypertext style of information access and retrieval is a *cognitive overloading* which could even lead the user to a loss of orientation. Although the term "context" is very generic, it is possible to identify in hypertext systems, some types of navigational contexts [SB95] as: arbitrary, topics derived, link derived, etc. These are explained later in this document. Contexts can be viewed from two different angles:

- an authoring context: enables the author to organise information into topics, themes or subjects, etc. This way, contexts are used to enforce intuitive authoring information classification activities.
- a user context: can be used to define presentation policies within a company, a group, etc.

Modelling Authoring Systems

A model offers an authoring language and makes development a structured task independent of the application. A major drawback identified in existing authoring systems is that they deal with the authoring of multimedia information and ignore the conceptual phase of a *project* design. This research considers both stages:

- *project authoring or authoring in the large:* where organisational and conceptual aspects of a project are considered. The project is viewed from a very different point of view. The semantics of the links among the elements of the project are referred to as *"application links"* since they differ from one application to the next;

- *authoring in the small:* illustrates the refinement and instantiation of the components of the project previously identified. This stage implements traditional authoring mechanisms such as synchronisation, media editing and linking, and presentation.

In order to present some of the aspects of the ACAM model, an example is used throughout this document. It consists of designing an authoring application describing the State of Pernambuco[3]. The application could start with a project authoring view that shows the main characteristics and subjects characterising this State. The presentation of which could be divided into several areas dealing with the Political, Economical, Historical, Cultural, Geographical and Touristical characteristics of this region of Brazil as shown in figure4. At the project level, a graph with *application links* can be used to show the way these components are related to each other. This mechanism was first used in the HDM model [GPS93]. An example of such a relation would be the link "events" that spans between the touristic and cultural information.

Authoring Paradigms

Authoring should be a systematic and a well structured process. There are very few techniques specifically designed for the authoring of hypermedia documents. A great deal of these are empirical techniques [GPS93]. Currently there is no completely "point-and-click" authoring system; some knowledge of algorithm design and heuristic thinking is required [Sig95]. As it is, authoring can be seen as an enhanced mode of programming where although the author neither has to learn a programming language nor has to use a specific Application Programming Interface (API), a basic understanding of how programs work is needed[4].

1. time and flow based iconic representation such as Authorware;
2. card based such as Apple's HyperCard;
3. frame paradigm;
4. script based systems such Hypertalk used in Apple's HyperCard and Toolbook from Asymetrix;
5. structure editing systems as described in the paper[LHB93];
6. frame direct timing relations paradigm used in Harmony[ea91] and Videobook[ROK90];
7. the cast score scripting paradigm;
8. collection based as in HM-Card [KAS94];

The ACAM user interface adopts adopts a combination of the flow icon based, collection and script paradigms with support for structure editing and run-time execution objects.

[3] Pernambuco is one of North Eastern states of Brazil

[4] A detailed description of existing paradigms is found in[dASeKLS95].

The ACAM Model

ACAM defines the four main modules shown in figure , namely, presentation, authoring, hypermedia and distribution.

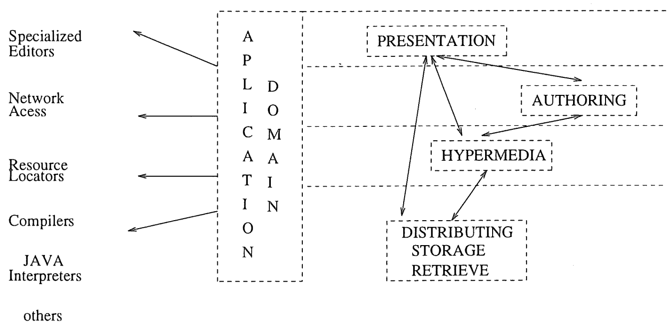

Figure 1: ACAM's Modules

Presentation

This layer executes the author's model designed during the authoring process, through the use of ACAM's mechanisms, services and the interaction with other tools and systems from its environment that are external to ACAM. Its main tasks include the execution of *pre-defined* synchronisation aspects, layout and content control, user interaction and event management. In order to cater for applications such as computer based training, (CBT) where it is often necessary for interaction between users, this module supports *distributed presentation scenarios.*

Authoring

Models the application, starting with a conceptual abstract model and refining this through the instantiation of its components. ACAM divides the authoring process into two stages: authoring in the large and authoring in the small.

Hypermedia

It is the hypermedia module that is responsible for the hypertextual aspects of the application. The following link types have been defined: application links, links of perspective, synchronisation links, conditional (flow control) links etc. This module is also responsible for the control and manipulation of the semantics associated with authoring contexts, version control, access control and security. Other services of this layer include the support of cooperative work and most of all the identification (object referencing) and transparent location of multimedia information objects.

With the emergence of new multimedia representation standards such as the *Multimedia and Hypermedia Expert Group* MHEG[MG95], already dealing with important multimedia issues including synchronisation, event manipulation, object identification, etc, there is no longer a need for a separate object layer. The hypermedia layer also handles links among objects that have semantics associated to their role in the model. Examples of these are synchronisation objects (e.g audio synchronisation object, etc), presentation objects, display objects, and monitoring objects.

Distribution, Storage and Retrieval

The new design architecture offered by the use of active *smart* [5] agents are being used instead of current rigid protocol based applications that lack flexibility and real distribution. Security concerns remain however, a major obstacle to overcome. A number of other platforms also support the use of re-usable and configurable active objects for the authoring of applications. [ea93b] and [ea93a] adopt the use of active objects. The MIT [ea94] model used in VuSystem, goes further in that it takes into account the distributed environment. VuSystm uses an extended TCL but has no specific support for user interaction. Examples of languages used by smart agents include Telescript, WAVE [Sap93], JAVA[Sun95b] etc.

Functional Architecture

Although an organisation of software modules into layers as the ones shown in figure , usually implies that one level makes use of the services of the layer below it, the layering shown here depicts a purely abstract hierarchy where higher levels are more abstract than lower ones.

The openness of the proposed model is shown with ACAM's integration to its environment. This integration is made with all the services offered externally. Such services would normally include, dedicated editors and presentation tools, communication software and tools such as WWW, FTP, WAIS, Java interpreters, MHEG engines, etc.

An Authoring Methodology

The design activity can be described as a two dimensional space between the author's mental model and the supported authoring methodology. Authoring oscillates between the two axis as shown in figure.

Figure 3 shows the four stages of the adopted authoring methodology:

[5] The term smart is used instead of intelligent to describe these agents in order to differentiate it from artificial intelligence based systems.

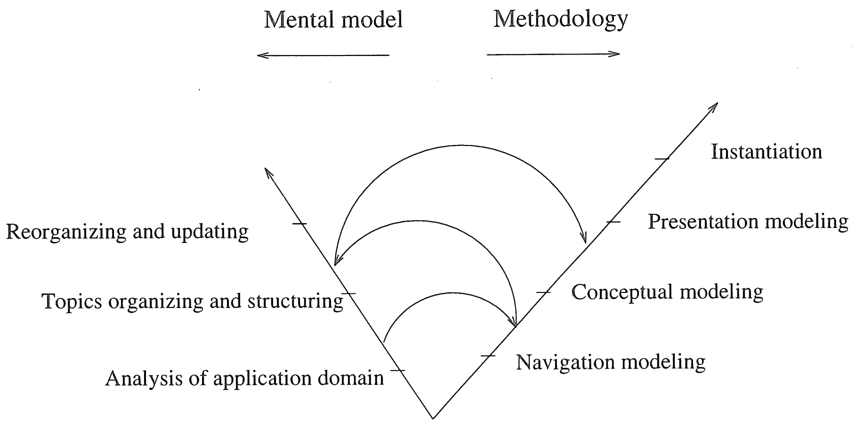

Figure 2: Authoring Design Activity

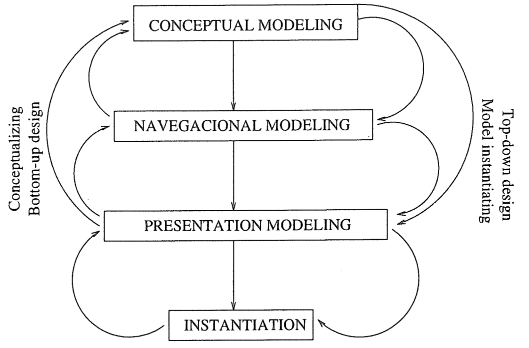

Figure 3: Hypermedia Applications Development Methodology

- conceptual modeling - an open authoring stage that allows the modelling of real world situations. It defines relations among topics, subjects or themes;
- navigational modeling - oriented towards the manipulation of the conceptual model previously created; it defines the navigational aspects through the promotion and election of parts of the topics, subjects or themes to the status of nodes; and the promotion or election of some of the relations to the status of links referred to as "application links";
- presentation modeling - at this stage is defined the presentation interface and user interaction aspects. Different aspects are considered, including static aspects such as application layout, dynamic aspects with synchronisation, and the behaviour of authoring objects under the effect of external user interface events;

- instantiation - multimedia data is created and associated to the elements of the presentation according to the pre-established author's model.

A consideration here, would be for the system to automatically extract or deduce the associated navigational and conceptual models. This part of the work is left for further research.

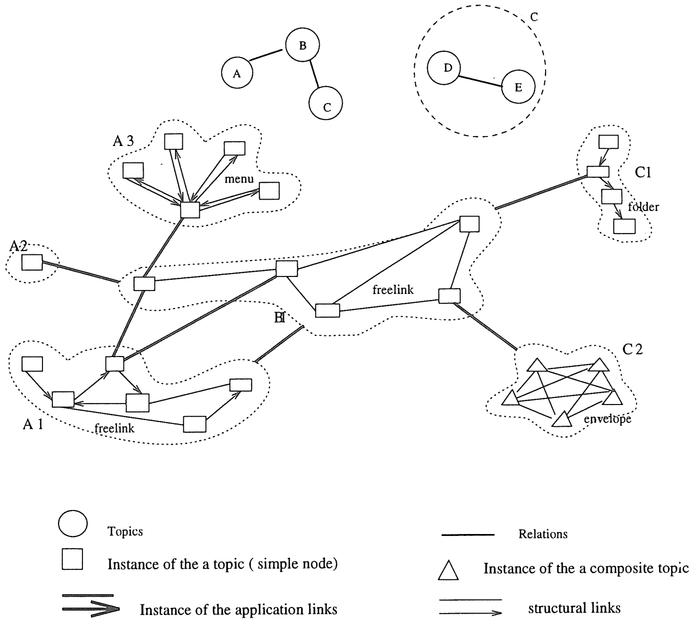

Figure 4: Hypothetical Conceptual Model

Figure 4 shows the conceptual model with the hypothetical topics: A, B and C. Where C, is considered a complex theme, which has sub-themes: D and E. Therefore topic C consists of an internal structure as well as content.

The topics (subjects or themes) shown in the figure, are considered as nodes in the second stage of authoring. They are represented by arcs in the conceptual model graph; these relations, which in the second stage could be application links showing relations among related topics and corresponding to different subjects, are represented using thicker lines. The content instances for a given topic are hyperdocuments rather than simple nodes. The reason behind this choice, is the ability to provide the author with complete freedom when instantiating the hyperdocument which can be a simple object, constructed by other nodes or complex with an associated navigational structure. Figure 4 shows that for each of the subjects treated there are few associated instances - A1, A2 and A3. These are instances of A; similarly B1 is an instance of B and C1 and C2 are instances of C. Each subject instance could be linked to other instances from other topics according to the pre-established abstract model. This way an applicative link between instances

A and C cannot be allowed since it is not specified in the model.

Structure of a Topic Node

An ACAM topic node consists of:

- *content:* information that characterises a topic, or more precisely, determines the behaviour of its instances such as: perspective links, security and access control constraints, structural navigational styles and other attributes.
- *internal structure:* used in order to represent complex themes subject to ZOOM-IN and ZOOM-OUT operations shown in figure5.

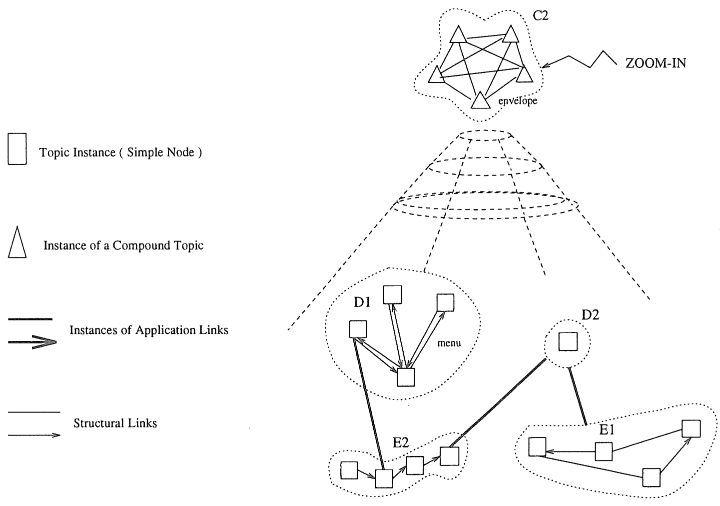

Figure 5: ACAM's use of Collections and Zoom Operations

Phase 1 - Authoring in the Large

Allows the creation of a model for a given specific application domain, see figure3. It defines the topics or themes, their relationships, attributes and operations; which are considered important during requirements analysis. In this initial stage, the domain application has been modelled as an abstract conceptual graph where vertexes and edges represent topics and their relationships respectively. In figure 4, the relationships among Tourism and Culture follows intuitively. This relationship permits a possible binding among related topics instances during the authoring in the small. Looking at the information system of the Pernambuco State, one sees that the themes are very generic. This way, one needs to create sub-themes that show more specific topics, but still belong to one or more of the abstract themes. Figure 6 shows the way ACAM handles topics and themes. Figure6 shows a conformance among the classification of the topics "Tourism and Culture" and the model. In other words, the author is required to follow the hierarchical structure shown in figure 6, and may further extend it with relationships among sub-topics.

Often the more abstract themes hold generic attributes, while simple ones have more specific characteristical information associated to them. The definition mode of this attributes and perspectives consist of:

- declaring the perspectives of the topics, e.g., in Geography we have text, graphics and audio;
- defining attributes identifying topics, e.g. the theme Tourism can be seen to contain the attributes: title, city-name, general-information and post-card;
- defining perspectives of the attributes which are sub-sets of the perspectives of the topics;
- not declaring anything, that is, make the model very abstract and as generic as possible;

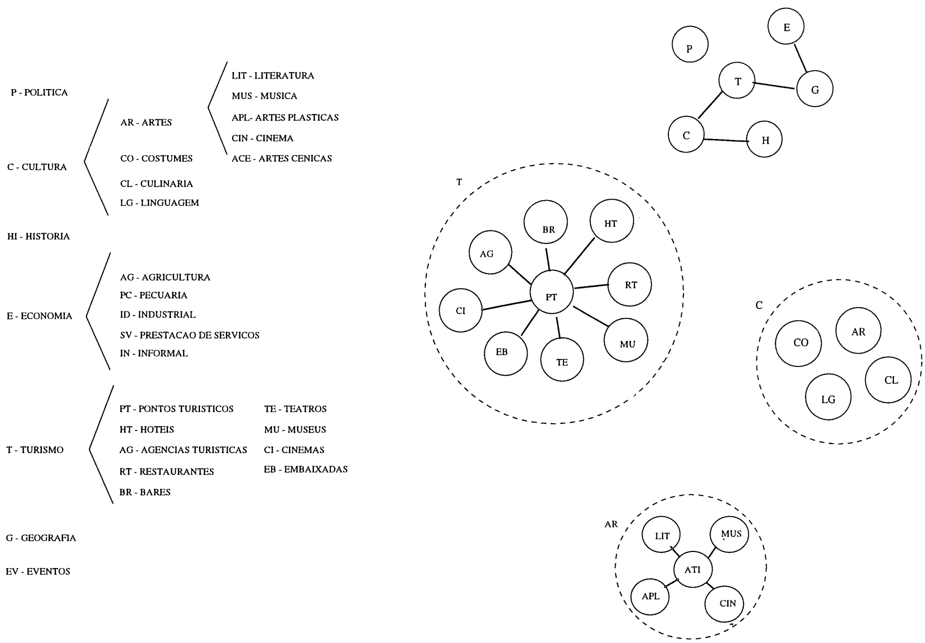

Figure 6: Practical Conceptual Model

Phase 2 - Navigational Modeling

Following the conceptual schema resulting from the previous phase, the author begins building a vision over the hypermedia system, creating the navigational basic objects based on the topics and relations defined in the conceptual modelling phase. Such objects include: nodes, links (arbitrary links, topics derived links, relation derived links and links derived from the internal structure).

Both conceptual and navigational modelling are going to guide the building of a specific multimedia authoring application. This new stage is known as authoring in the small within ACAM. It defines the final stages of the authoring process responsible for modelling the presentation and instantiating application information components.

Phase 3 - Presentation Modelling

The presentation layer deals with issues such as the application layout and the synchronisation of objects built during the instantiation stage. This third stage of the authoring process models the application presentation aspects in time and space. The following describes the authoring aspects of presentation modelling:

- *static aspects:* that deal with the application object layout information. The model adopts the use of user definable templates as background guidelines for all or part of the application objects and topics.
- *dynamic aspects:* represent presentation functions that deal with the synchronisation of multimedia objects built during the instantiation phase.

Other dynamic presentation aspects include support for the dynamic adaptation to resource availability that can be provided by invoking methods on associated objects at run time. This mechanism allows the building of different presentation policies and schemes. CMIF[Bul91] is an example of a system that adopts this mechanism where each component can be represented by a number of media (such as video versus audio and text, etc). The presentation layer selects which option to use depending on resource availability during run-time.

The authoring model ACAM defines a number of related editors that control how the authoring paradigms can be manipulated to suit the author's specific application:

- *time based flow editor:* used to offer a hierarchical synchronisation service;
- *a temporal editor:* responsible for the specification of sophisticated inter-media synchronisation;
- *script editor:* programs the events declared in the flow editor.
- *layout editor:* is responsible for the definition and manipulation of object layout and formatting aspects;
- *content editors:* represent all media dependent dedicated editors such text, bitmap, graphics, and other editors.

Phase 4 - Instantiation

It is responsible for the creation of multimedia data and its association to the elements of the presentation according to the pre-established author's model and templates. ACAM defines different types of links including navigational, of perspective, application links, etc. These have the structure defined by the attributes: identifier, type, name, entity of origin, entity of end, symmetry, method, etc. Access structures are used to define the way a user may access the authored information. These include: routes, indexes, user queries and associative access.

Conclusion and Future Work

This paper presented a hypermedia authoring model based on the concept of context based authoring and which uses underlying communicating agents to provide adaptive run-time reference resolution, access and loading of information and software applications, and distributed processing. Although distributed platforms have been among us for a long time now, they have received limited commercial support. It is the authors view that the future lies in the use of smart agents and languages for building distributed applications.

With the distribution of the whole model a number of advantages have been achieved including load balancing, support for users with limited resources such as mobile users, redundancy of the components which increases system availability, etc.The adoption of the Java language brings a number of advantages as far as Internet access is concerned. The author may dynamically load referenced external objects using this capability supported by the Java language during run-time. Last but not least, an *interactive* demonstration of the prototype will be made available through the Internet to remote users. This may be accessed through Internet browsers that support a Java interpreter such as the netscape WWW browser and the HotJava[Sun95a] browser from Sun MicroSystems.

One of the positive aspects of ACAM is that is has been designed to be generic enough to capture a number of applications and systems, to include new functionality and modules as these are identified and its openness through the run-time interpretation of Java code. There are, however a number of remaining open issues that one would need to see them dealt with. An important one is the management and use of Quality of Service (QoS) controls. Although, the model supports a high level adaptive use of the environment through the definition of the concept of perspectives, there is no explicit use made of QoS parameters.

Support for cooperative authoring is another aspect of authoring that is lacking from the current model. This, clearly is an important characteristic and needs further consideration in the coming stages of this project.

Acknowledgements

This work has been undertaken as part of the Project SAM (Sistemas de Autoria Multimedia), sponsorship by the CNPq/Brasil of the ProTem-II program.

References

[All95] Michael W. Allen. *Authorware Models for Instructional Design.* Prentice-Hall, 1995.

[BB95] John Bates and Jean Beacon. Supporting interactive presentation for distributed multimedia applications. *Multimedia Tools and Applications - An International Journal*, 1(1):47–78, March 1995.

[Bul91] D. Bulterman. A structure for transportable dynamic multimedia documents. In *USENIX Multimedia for Now and the Future*, 1991.

[dASeKLS95] Ricardo Martins de A. Silva e Kleber Leal Sales. Um estudo sobre modelos e sistemas de autoria em multim/'idia e hipermídia. Technical report, Departamento de informática da Universidade Federal de Pernambuco, 1995.

[ea91] K. Fujikawa et al. Multimedia presentation system harmon with temporal and active media. In *USENIX: Multimedia for Now and the Future*, pages 1–10, 1991.

[ea93a] S. Gibbs et al. Application construction and component design in an bject oriented multimedia framework. *Visual Objects - University of Geneva*, 1993.

[ea93b] S. Gibbs et al. Data modelling of time-based media. *Visual Objects - University of Geneva*, 1993.

[ea94] D. Tennenhouse et al. Software oriented approach to the design of media processing environments. In *Proceedings of the first IEEE International Conference on Multimedia Computing and Systems*, 1994.

[GPS91] Franca Garzotto, Paolo Paolini, and Daniel Schwabe. HDM - A Model for the Design of Hypertext Applications. *Proceedings of Third ACM Conference on Hypertext*, pages 313–328, December 1991.

[GPS93] Franca Garzotto, Paolo Paolini, and Daniel Schwabe. HDM - A model approach to hypertext application design. *ACM Transactions on Information Systems*, 1(11):1–26, January 1993.

[IMA93] IMA. Multimedia system services. Technical report, nteractive Multimedia Association, 1993. Version 2.0.

[ISO92] ISO-100744. Information processing – text and office systems – hypermedia/time-based structuring language — (HyTime). International Standard IS-100744, International Standards Organisation, Geneva, 1992.

[KAS94] Hermann Maurer Keith Andrews and Nick Scherbakov. Semantic modeling in hypermedia: An object-oriented approach. Technical report, Institute for Information and Computer Support New Media, Graz University of Technology, 1994.

[LHB93] G. van Rossan L. Hardman and D. C. A. Bulterman. Structured multimedia authoring. In *Proceedings of the first ACM Conference on Multimedia*, pages 283–289, 1993.

[Mac90] MacroMind. *Dynamic Media Authoring Tool for the Apple Macintosh Director Version 2.0.* Macromind, 1990.

[MG95] Multimedia and Hypermedia Expert Group. Information technology - coding of multimedia and hypermedia information- part 1. Technical report, ISO/IEC, 1995.

[NO94] P. Newhouse and R. Oliver. *Authorware 2: An Introduction to Interactive Multimedia with Authorware Professional 2.* 1994.

[Pei95] Mauro Sergio Oliveira Peixoto. Modelagem de sistemas de hipertexto: Paradigmas e uma aplicação prática real. Master's thesis, Departamento de InformáUniversidade Federal de Pernambuco, July 1995.

[ROK90] H. Harada Ryichu Ogawa and A. Kaneko. Scenario based hypermedia: A model and a system. pages 38–51, November 1990.

[Sap93] Peter Sapaty. A brief intoduction to the wavelanguage. Technical report, Universitat Karlsruhe, February 1993.

[SB95] Daniel Schwabe and Simone D. J. Barbosa. Navigation modelling in hypermedia applications, 1995.

[Sig95] J. Siglar. Multimedia authoring systems frequently asked question (faq). Technical report, The Internet, 1995.

[Sun95a] Sun. The hotjava browser: A white paper. Technical report, Sun Microsystems Computer Corporation, 1995.

[Sun95b] Sun. The java language specification. Technical report, Sun Microsystems Computer Corporation, 1995.

[VM93] M. Vazirgiannis and C. Mourlas. An object oriented model for interactive multimedia presentations. *The Computer Journal*, 36(1), 1993.

[Wan93] P. Wanyer. Inside quicktime. *Byte*, 16(13), 1993.