MULTIMEDIA AUTHORING SYSTEMS: A PROPOSAL FOR A REFERENCE MODEL

Mario Lorenz (Mario.Lorenz@informatik.tu-chemnitz,de)
René Schmalfuß (Rene.Schmalfuss@informatik.tu-chemnitz.de)

Chemnitz University of Technology
Computer Science Department
09107 Chemnitz
Germany

ABSTRACT

This paper describes a proposal for a reference model for multimedia authoring systems. In this context a multimedia authoring system is a combination of hard- and software for making, presenting/running and revising the entire multimedia document and its informations. The reference model framework consists of the user component, the knowledge component, the design support component and the system component. The framework specification was done top down in a modular way and it includes support for various design philosophies, design workgroups, networking and document lifetime cycle models.

INTRODUCTION

Today the term Multimedia is associated with a large amount of applications. The spectrum of applications covers the area from simple dialog style communication systems to complex networked information presentation and retrieval systems. Observing the increasing number and complexity of applications the need for powerful specialized authoring tools is visible. On the other hand there is a need for bulit in system compatibility and information exchange within the authoring process. Currently most authoring systems provide support for a concrete document design philosophy. Therefore it is very difficult or impossible to exchange parts of or entire documents between various systems and platforms. Multimedia documents designed with classical authoring systems typically can only be viewed with system specific so called runtime modules. Mostly the information exchange interface of authoring systems is implemented as import or export of multimedia raw data only.

This proposal for a reference model is oriented towards all systems are used to produce and revice multimedia documents in any matter. Our goal is to provide a vision how the authoring systems should be designed and how they should work. Users should be able to classify and grade existing systems with the help of our reference model.

BASIC NOTATIONS

A **multimedia authoring system** is designed for making and revising of multimedia documents. It consists of hard- and software components for producing, importing and editing multimedia raw data and combining them based on a specific design philosophy.

A **multimedia document** is a structured quantity of multimedia raw data in a specific format. It may contain time dependent and time intepedent informations and may be used interactive or as a self running application. There are two basic structures characterising the document: the static and the dynamic structure.

The **static structure**, in interactive useable documents sometimes called the navigation structure, describes the relations between the informational subsets, means the topology of the document. The **dynamic structure** additionally describes the temporal aspect of the informational flow within a document. The definition of the dynamic structure is necessary if the document contains time dependent informations or if the document should be used as a self running application.

The **design philosophy** describes how to combine informations to the document. There are three classical philosophies in multimedia authoring :

- the **frame oriented** design,
- the **timeline oriented** design,
- the **flowchart oriented** design.

Because each of these philosophies has its advantages and disadvantages many practical authoring systems use combinations or enlargements of these philisophies. A typical extension is a timeline philisophy with a integrated programming language, which brings the functionality needed to build hyper based documents.

DEMANDS ON THE REFERENCE MODEL

A future multimedia authoring system should support the entire process of making and revicing a multimedia document by its functionality. This means it has to support all direct and indirect tasks within this complex process as good as possible. Aside from the essential design and implementation tasks this includes project management tasks, the coordination of cooperative work in workgroups and tasks regarding document revision and update. It is apparent from this that the needed hugh amount of funtions is very complex to describe.

Therefore we considered some guidelines within the design process of this proposal :

- **modularization :** A practial authoring system is typically large sized and very complex. It is necessary and possible to divide the entire system into seperate parts or functional blocks which are idealy logical distinct. The reference model was specified top down in a modular way to provide clarity of its concepts.

- **various degrees of abstraction :** One goal of this reference model is to show the basic concepts of multimedia authoring systems. On the other hand the model should describe concrete design and functional aspects. Together with modularization and the top down design the various degrees of abstraction are given.

- **life cycle concepts :** From software technology the importance of design and life cycle concepts, e.g. the waterfall or the prototyping model, is well known.

  Because the authoring process of multimedia documents is compareable with the making of software, there should be a built in or configureable support for various life cycle concepts by the reference model.

This means the model has to have a design support component which integrates the necessary functionality, e.g. project management and documentation.

- **team work management :** Of course the authoring process is fit together from various subprocesses of many disciplines, e.g. audio and video editing, graphics design or frame layouting. A good looking and well functional multimedia document requires the assistance of specialized people.

  Because the design tasks may be very expensive there is a need for workgroup management. The reference model should support basic concepts of workgrouping, e.g. concepts known from CSCW (computer supported cooperative work) [Syrin93a].

- **openess to other models :** The making of multimedia documents is a complex process consisting of subtasks which have been discovered by other disciplines.

  Therefore the model should be open to be combined with the concerning reference models or standards of these disciplines or the model should integrate these ones.

  For example, the focus is on the main standards like the Computer Graphics Reference Model [ISO92a], the Amsterdam Hypertext Model [Hardm94a], software lifecycle models, the Seeheim Model [Pfaff85a] for user interfaces and the OSI reference model [Stött89a] for networking.

  Concepts of all these and more models may influence the authoring process of a multimedia document and therefore these concepts should be noticed in this proposal.

OUTLINE OF THE REFERENCE MODEL

Like in some other models, e.g. CAD reference model [Abeln95a], the core of the reference model is a generic architecture. In this case it describes a generic multimedia authoring systems. This architecture consists of the following parts :

- the **user part** : The set of all application specific components available within the authoring system to implement the functions needed for the concerning design philosophy.
- the **system part** : The set of all application independent components available within the authoring system are needed to support the functioning of the user part components.
- the **design part** : The unity of design philosophy and design data to describe a class of multimedia documents and informations regarding their life cycle.
- the **knowledge part** : The design philosophy specific knowledge of making a multimedia document described within the design process.

Dividing the entire architecture in design philosophy dependent and independent parts allows the exchange of parts and therefore the configuration of the model due to the design philosophy. This and the used modularisation is needed to enclose a wide range of systems with various design philosophies.

Each component of the reference architecure is described by the following four characteristics :

- **component substructure** :
  parts contained in the component.
- **component dynamic** :
  internal functions of the component.
- **component interface** :
  communication with other components.
- **component data** :
  values of the component internal variables.

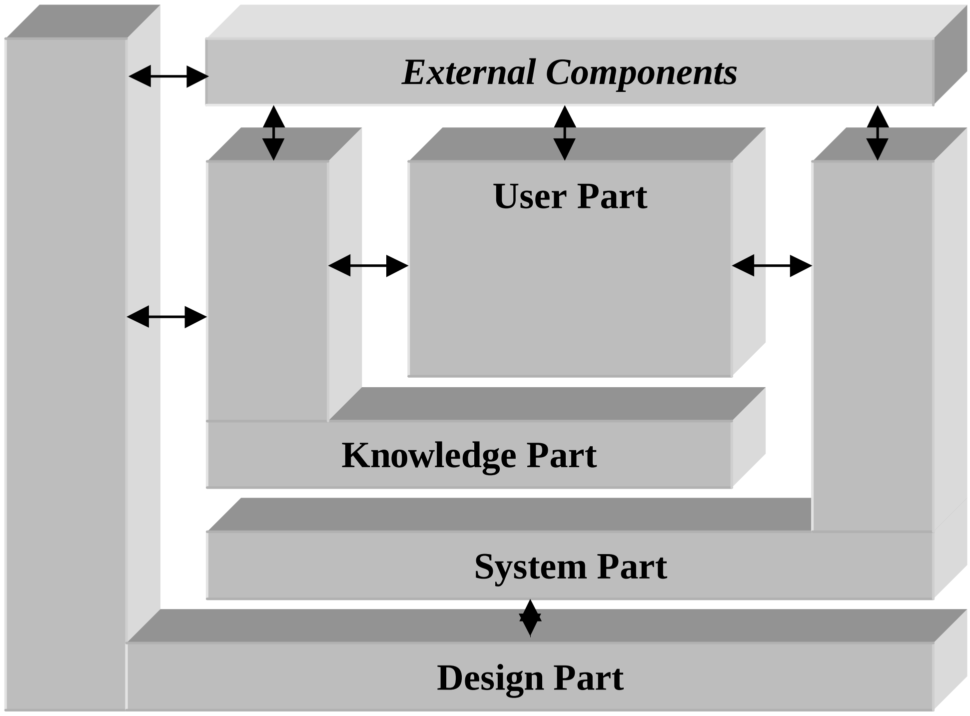

Fig. 1 : framework of the reference architecture

User part

The user part contains all components of the authoring system needed to build the multimedia document. In this context the components are application specific.

Based on the used design philosophy it is possible to subdivide the user part into three layers :

- the **philosophy independent layer** : The basic functionality for handling raw multimedia objects, such as editors, grabbers, import or export filters and production tools**.**

- the **user specific layer** : The advanced functionality used to handle and process multimedia objects, such as data converters, filters and compression codecs and tools for information processing.

- the **philosophy specific layer**: The functionality needed to combine the reviced multimedia objects to a document regarding the concrete design philosophy, such as structure and script editors, procedural control tools and verification aids.

The consequence by this subdivision is that the higher, philosophy dependent functionality is based on the lower one. Therefore it is possible to configure the user part for a concrete design philosophy by exchanging the tools located in the higher layers.

Whereas the specific of the functionality increases in higher layers, the general applicability increases in the lower layers.

This causes a functional hierarchy within the user part. Any functionality in a layer can only be used by functions of this or of higher layers.

The user part has four logical interfaces :

- the **knowledge retrieval interface** for calling context based and user specific informations.
- the **management interface** for communication with the design part concerning informations such as analysis, specification, documentation and team work management data.
- the **operating interface** for communication with the system part.
- the **external interface** for communication with external tools.

The operating interface is the only one which is physically implemented. This is caused by the demand that all communication within the reference architecture occur over one communication channel located in the system part.

System part

Seen from the viewpoint of implementation the system part represents the lowest layer of the authoring system. It implements all system specific tasks needed to run any functionality within any part of the reference architecture. Its main task is to act as a interface between the basic operating system and the higher parts of the authoring system. As described above the system part also supports the system configuration, management and communication tasks.

It provides the necessary functionality for the following tasks :

- the **system ressources** to implement and control the basic system funtionality such as information storage and recovery, basic dialog tools and tools for handling raw multimedia data.
- the **configuration manager** to setup the system for a concrete design philosophy and user demands
- the **communication manager** to handle the internal communication between the system parts in form of a message based communication pipeline.

- the **interface manager** to control the system communication with local and remote users containing the information presentation, the user interaction and the system access through a network port.

- **user manager** to support the user and workgroup administration tasks within the design process and to realize the user session management tasks.

Locical interfaces are defined to explain and clearify the information flow handled by the system part:

- the **operating interface** describes the system internal communication with the user part, the design part and the knowledge part. It is implemented as a message based communication pipeline.
- the **user interface** describes the all components and tasks to realize the interaction with local users including desktop information presentation, dialog handling and access control.
- the **network interface** is intented to specify the information exchange between networked systems including system access control, the transfer of raw multimedia data, the synchronization of shared access to raw data or document parts and workgroup management.

Seen from the viewpoint of configuration the system part itself is partial configureable by its configuration manager. This makes it possible to fit the reference architecture to the concrete needs concerning e.g. raw multimedia data formats, network protocols or workgroup structure. On the other hand there are basic components of the user part that cannot be configured an any way, e.g. the system ressource management or the internal communication pipeline.

Design part

To support the user or the workgroup within the complex design and revice tasks of multimedia documents some life cycle models known from software technology were examined regarding their applicability in this context. The specification of the design component was fundamental influenced by the known design philosophies named above and the results of our examinations.

Therefore the design part contains the needed tools for the concerning design philosophy :

- **project management including analysis, specification, design, test, maintenace and document revicing**
- **team work planning and management**
- **documentation tasks.**

The design part is configureable for the needs of a concrete design philosophy. This causes the adaption of a life cycle model by the choice and setup of available components.

For example, if the design task is the making of a hyper media based flow chart oriented document for presentation of a new product by prototyping, the configuration should include the existence of a runable document framework with a minimal functionality.

The design part is connected to the user part and the system part by the management and operating interfaces. Both interfaces are shown above.

Additionaly the design part has a **external interface**. This is intended for integration of external tools, e.g. a documentation system or a existing CASE-Tool.

Knowledge part

This part is designed to support the users with the knowledge needed to design multimedia documents under the use of the authoring system in a effective way.

This knowledge can be subdivided into :

- **problem independent knowledge :** informations on handling and processing of raw multimedia data, data formats and regarding standards, media characteristics, media handling, ...

- **problem dependent knowledge :** informations regarding the design philosophy, e.g. design characteristics, ergonomic guidelines, representation parameters, regarding life cycle models, ...

- **user specific knowledge :** informations about users role in the design process, e.g. workgroup organization, user specific guidelines or standards, user specific characteristics of document design, ...

Present authoring systems have none or a minimal knowledge component in form of a help system. The users have to retrieve informations concerning the design context by the use of external media or applications.

Because the knowledge is divisible in formal and informal knowledge, it is not planned to fill the knowledge component with a hugh amount of informations but with effective algorithms for retrieval of informal and formal knowlegde of mostly external sources.

Therefore the knowledge part has additional to the knowledge retrieval and operating interfaces a external one.

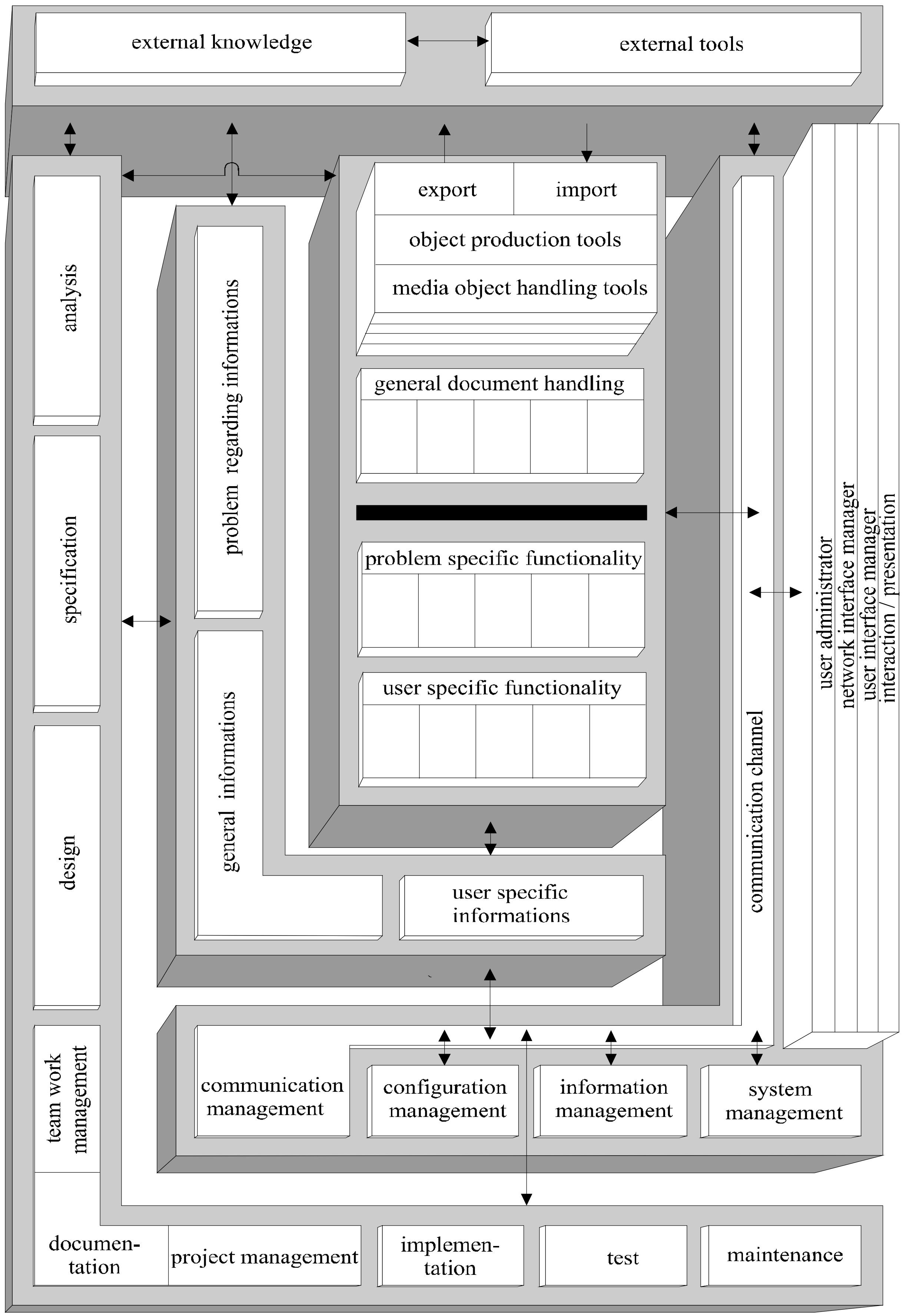

Fig. 2 : Framework Specification of the reference architecture

SUMMARY

The paper describes some topics of our research in form of a proposal for a reference model for multimedia authoring systems. The ongoing activities are concentrated in the consecutive detail specification of the functional aspects of the four architecture parts and their subcomponents. Fig. 2 shows the framework specification of the reference model architecture core.

REFERENCES

[Abeln95a] O. Abeln: CAD-Referenz-modell. Teubner Verlag, 1995

[Hardm94a] L. Hardman, a.o. : The Amsterdam Hypermedia Model. ACM,Vol.37, No. 2, 1994

[ISO92a] Computer Grahics Reference Model. ISO/IEC IS 11072, 1992

[Pfaff85a] G.E. Pfaff: User Interface Management Systems. The Seeheim Workshop. Springer Verlag, 1985

[Stött89a] K. Stöttinger: Das OSI Referenzmodell. Datacom Buchverlag, 1989

[Syrin93a] M. Syring: CSCW - Konzepte, Methoden und Anwendungen. Addison Wesley, 1993.