Josef Kolbitsch

Aspects of Knowledge Management

Master's Thesis in Telematics

Dr. Hermann Maurer
Institute for Information Processing and Computer-Supported New Media
Graz University of Technology

Dr. Jennifer A. Lennon
Hypermedia Unit, Department of Computer Science
University of Auckland

Auckland, New Zealand, September 2002 – July 2003.

Josef Kolbitsch

Aspekte des Wissensmanagements

Diplomarbeit in Telematik

o.Univ.-Prof. Dr.phil. Dr.hc.mult. Hermann Maurer (Begutachter)
Institut für Informationsverarbeitung und computergestützte neue Medien
Technische Universität Graz

Dr. Jennifer A. Lennon (Betreuerin)
Hypermedia Unit, Department of Computer Science
University of Auckland

Diese Arbeit ist in englischer Sprache verfaßt.

Auckland, Neuseeland, September 2002 – Juli 2003.

Preamble

This thesis is submitted in fulfilment of the requirements for the degree of Master of Sciences ("Diplomingenieur der Telematik") at Graz University of Technology.

Ich erkläre an Eides statt, daß ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe oder unerlaubte Hilfsmittel verfaßt, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Weiters versichere ich, dieses Diplomarbeitsthema bisher weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt zu haben.

Acknowledgments

My thanks go to Prof. Maurer for his support in my endeavour to write my thesis at the University of Auckland. I would like to thank Jennifer Lennon very much for her invitation to Auckland, for her very warm welcome, her friendship and encouragement, her help, professional advice, and support. She introduced me to scientific writing … and I tried to understand and implement it. Thanks also to all other people at the University of Auckland who helped me.

I would like to thank Richman for his friendship, for his tolerance of my music, our dinner experiences, the numerous talks and discussions, and for his help with English language problems when I needed it.

Thanks to my friends who visited me in New Zealand. You know how much i liked and appreciated it! Thanks also to all my friends who have, for one reason or another, not visited me; thanks to them for remembering me, for still being my friends and welcoming me back to Graz. A very big thank you to my friends who were willing to have very long phone calls with me, who stayed up late to chat with me on the internet, and who spend hours e-mails to me. Thank you to all who helped and supported me – you know who you are.

Thanks to my parents for their support, love, encouragement, and patience during the last few years at university. Thanks, of course, also to my extended family.

To my parents.

And to my grandfather, with whom I have always wanted to go to Sweden.

Abstract

Active documents are a technology in knowledge management that allows users to ask questions to documents and enables them to communicate with the content. In combination with other techniques, a knowledge management system can be designed that operates proactively for, and interactively with, the user.

In this thesis, the basic active documents concept is extended to work with documents of any type of media. Based on these foundations, two systems are proposed: ADIME and VIVID.

ADIME is an environment for active documents in medical education. It introduces features of knowledge management in traditional, yet media-rich e-learning systems in the medical domain. One of the key features is that it allows users to ask questions or add annotations to any media object in documents such as images and video clips.

VIVID, the second system proposal, is an elaborate concept for enhanced interactivity in digital video broadcasting environments. It improves the conventional broadcasting paradigm in that it permits users to find answers to content-related questions while they are actually watching a programme. This makes it easier for consumers to find, for instance, names of actors or the title of a sound track. Moreover, VIVID can provide functions for users to extract certain segments or objects from the content. This feature is particularly useful for downloading a short video clip or obtaining the audio track of a broadcast.

The described concept can be employed with a wide variety of content types including movies, news and documentaries, sports, music television, and children's programmes. The range of possible client devices includes integrated digital television sets and set-top boxes, networked computers, and mobile devices such as handheld computers or mobile phones.

Keywords: Knowledge Management, Active Documents, Metadata, Digital Rights Management, Digital Video Broadcasting (DVB), Digital Television, Interactive Television, e-Learning.

Kurzfassung

Aktive Dokumente sind eine im Wissensmanagement verwendete Technik, die es Benutzern erlaubt, Fragen an Dokumente zu stellen und mit Dokumenten zu kommunizieren. In Verbindung mit anderen Methoden kann ein Wissensmanagementsystem entworfen werden, das proaktiv für und interaktiv mit dem Benutzer arbeitet.

In dieser Diplomarbeit wird das fundamentale Konzept von aktiven Dokumenten erweitert, sodaß es mit Dokumenten jeglichen Types verwendet werden kann. Aufbauend auf diesen Grundlagen werden zwei Systeme konzipiert und vorgestellt: ADIME und VIVID.

ADIME ist eine Anwendung aktiver Dokumente in der Ausbildung im medizinischen Bereich. Mit ADIME werden Funktionen aus dem Umfeld des Wissensmanagement in traditionellen, jedoch multimedial aufbereiteten, medizinischen e-Learning Systemen eingeführt. Ein Charakteristikum ist, daß Benutzer Anmerkungen an beliebige multimediale Objekte wie etwa Video-Clips of Bilder anfügen und Fragen an diese multimedialen Objekte stellen können.

VIVID, der zweite Entwurf, ist ein komplexes Konzept für verbesserte Interaktivität in der Domäne der digitalen Videosendungen. Es erweitert das herkömmliche Paradigma von Videoausstrahlungen, indem es Benutzern ermöglicht, Antworten auf inhaltsbezogene Fragen zu finden, während sie ein Programm ansehen. Dadurch ist es für Zuseher einfacher beispielsweise Namen von Schauspielern oder den Titel eines im Hintergrund gespielten Liedes herauszufinden. Weiters kann VIVID Benutzern Funktionen anbieten, die es gestatten, Segmente oder Objekte aus einer Sendung zu extrahieren. Dies ist besonders nützlich, um etwa einen kurzen Video-Clip oder den Soundtrack eines Filmes erwerben.

Das beschriebene Konzept kann mit einer breiten Palette von verschiedenem Videomaterial verwendet werden: mit Filmen, Nachrichtensendungen und Dokumentationen, mit Sportübertragungen, Musiksendungen oder aber auch Kinderprogrammen. Mögliche Endgeräte für Benutzer schließen digitale Fernsehgeräte und Computer mit Netzwerkverbingunden genauso ein wie Mobiltelephone, Handheld Computer und andere mobile Geräte.

Stichwörter: Knowledge Management, Active Documents, Metadata, Digital Rights Management, Digital Video Broadcasting (DVB), Digital Television, Interactive Television, e-Learning.

Contents

Part 2: Extensions to Active Documents

Part 3: Active Digital Video Broadcasting

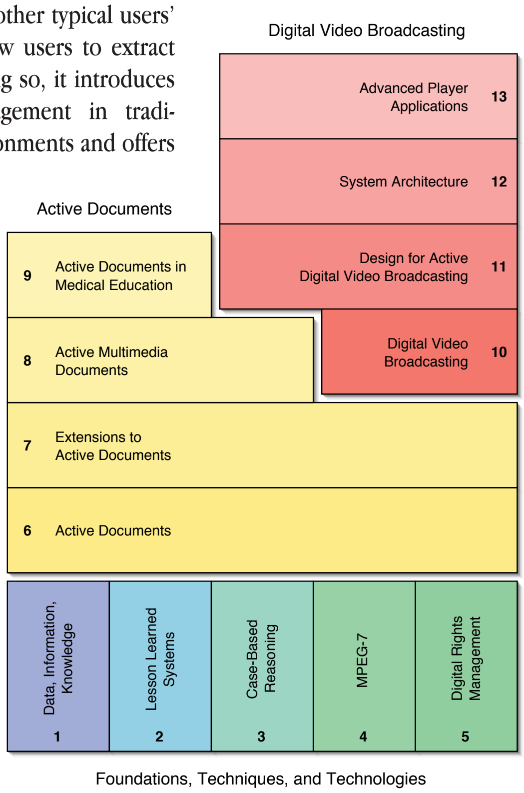
Appendices

Introduction

Do questions such as *"In which movie have I seen this actress before?"* of *"What is the title of the song in the background?"* sound familiar to you? Today's digital video systems including DVDs, digital television (DTV), and computer-based video players are not capable of answering these questions.

The main part of this thesis proposes Vivid, a system for Virtual Interactivity in Video broadcasting environments. Vivid makes it possible to provide answers for the questions above and for many other typical users' requests. Moreover, it can allow users to extract and download content. In doing so, it introduces features of knowledge management in traditional video broadcasting environments and offers interactivity to consumers.

The proposed system makes use of latest techniques for describing content and digital rights. It can be implemented for use with a wide variety of client devices including DTV, conventional and wireless computing networks, and ubiquitous computing devices such as mobile phones or handheld computers.



Usage Scenarios

Vivid can be employed with a range of different content types including movies, news and documentaries, sports, and children's television. Two scenarios are outlined in the following paragraphs.

Movies. A person is watching a movie on TV and wonders where he has seen this particular actress before. The viewer uses a pointer device built into the remote control and selects the actress on the screen. A window pops up and presents information about the actress: her name and role in the movie, her real name, and other data. In addition to this, a reference to a specialized movie database such as the Internet Movie Database (IMDb) is provided, so the user can obtain a list of all movies in which the actress occurred. In a similar way, the viewer can find out which other movies the director has directed, who has written the book or script, whether the film has been awarded, etc.

Another aspect focuses on the sound track of the movie: users like a particular song and would like to have it. They request information on this scene, a window on the screen pops up and lists all noteworthy elements of the scene. The list includes the names of all actors and also all audio clips in the scene. Now, the user knows the name of the song and the artist, and can also choose to download it instantly to the set-top box, computer, etc.

News and Documentaries. News and documentaries can sometimes be quite difficult to follow if one is not familiar with the context, domain specific facts, or the historical background. In this case, references to related "articles" or background information can be very useful. Basically, this is an analogy to many news services found on the WWW.

In a news broadcast, VIVID can be used to offer more specific details for interested viewers. A news item about the Columbia space shuttle disaster, for instance, can be supplemented with a graph showing the flight path, a chronology of the accident, related news stories, etc. A hyperlink to the Nasa website complements the report.

Documentaries can use similar features: imagine a school class watching a film about World War II. The teacher wants to explain some events in detail and therefore she requests additional information from the system. A variety of audio documents, other video clips, maps, text documents, etc. are readily available. The teacher selects a historical map, and has it displayed on the screen. Alternatively she could download it and print it out.

Technical Aspects

The idea of VIVID is based on a variant of active documents where answers to frequently asked questions are provided proactively without the user explicitly requesting them.

From a technical perspective, the functionality of the system is provided through a sophisticated application of metadata and digital rights. Every scene of a movie, for example, is supplemented with a rich set of metadata that contains information such as the names of all actors appearing in the scene, the name of the sound track and possibly a reference to it. Media objects also have a digital rights record attached, which determines if certain parts of the content such as the sound track may be extracted and downloaded.

Standards employed by the proposed system include MPEG-7 for metadata and MPEG-21 for digital rights. The content can be encoded using MPEG-2 (television environments), MPEG-4 (computer-based and mobile devices), or similar technologies.

The transmission of content and digital data relies on diverse technologies including the Digital Video Broadcasting Specification (DVB) for digital television equipment, MPEG transport streams over conventional Ethernet or ATM computer networks, and cellular networks such as GPRS or UMTS for mobile devices.

Organization of the Thesis

This thesis, *Aspects of Knowledge Management,* is subdivided into three parts. The first part provides the foundations for later chapters by introducing the techniques and concepts on which the research in this paper is based.

In the first chapter, the meaning of data, information, and knowledge is clarified, and the Maurer-Tochtermann-Model as an elaborate model of a KM-system is presented. Lesson Learned Systems, a practical application of KM in large organizations, are described in the subsequent chapter.

One of the techniques frequently employed in KM is case-based reasoning (CBR). The CBR approach as a problem solving method is briefly explained in chapter 3. State of the art KM-systems also utilize technologies such as metadata descriptors and digital rights management in order to facilitate the use of multimedia documents. Moreover, their use enables more sophisticated features such as content-based retrieval. Chapter 4 introduces MPEG-7 as one of the most recent metadata standards. Digital rights management is characterized in the following chapter.

After having established the basic requisites, the remaining two parts of the thesis address two destinct aspects of knowledge management: active documents and active digital video broadcasting.

As an introduction to Part 2, Maurer's notion of active documents is described. Chapter 7 provides several examples of how active documents are used together with complementary techniques in digital libraries and knowledge management systems. In chapter 8, the elementary concept of active documents is extended so that it can be used in the multimedia domain with objects such as images, sound, or video content.

Part 2 culminates in a proposal for ADIME, Active Documents in Medical Education. ADIME is an application of active multimedia documents that is targeted for the use in the education of students at medical schools.

The third part of this thesis deals with a concept for active digital video broadcasting that allows the user to easily access content related information. The first chapter in this section presents an overview of current digital video initiatives and corresponding standards. Based on this introduction, the elementary idea of active digital video broadcasting and VIVID, a system for Virtual Interactivity in Video broadcasting environments, is explained.

Subsequently, the basic architecture of the proposed system is illustrated, and the features and requirements of two prototypes of advanced player applications for VIVID are specified.

At the end of the thesis the appendices including an essay about the convergence of television and the internet, an MPEG-4 video compression test, a poster, a short research report, and presentation slides for Part 3, as well as the references section, and an enhanced index can be found.

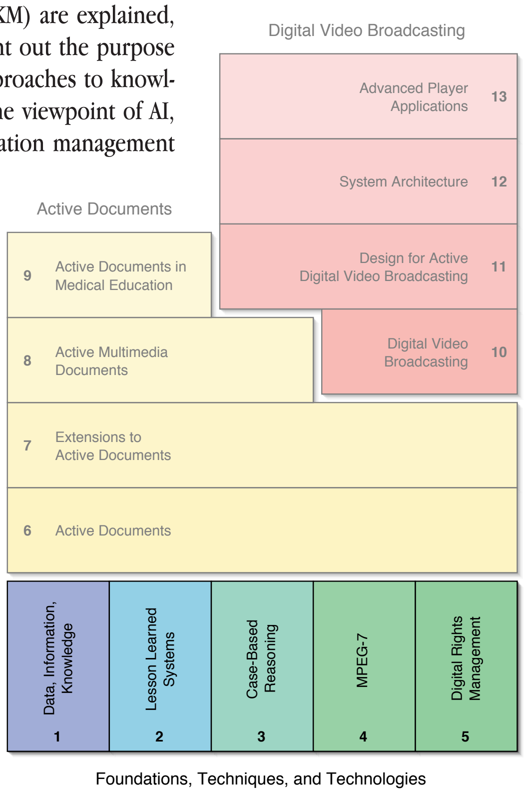
Part 1

Introduction to Knowledge Management

Introduction to Knowledge Management

The first part of this thesis is a general introduction to knowledge management. As such it provides the underpinning technologies and techniques that are used throughout Part 2 and Part 3.

Chapter 1 defines the terms data, information, and knowledge, and describes the differences and their relationships among each other. Based on these definitions, the contrasting types of knowledge-based systems including knowledge engineering, knowledge processing, and knowledge management (KM) are explained, and an attempt is made to point out the purpose of KM. After discussing two approaches to knowledge management, one from the viewpoint of AI, the other one from the information management perspective, the Maurer-Tochtermann-Model of knowledge management is presented as an example for the sophisticated architecture of a KM-system.



The second chapter deals with a concrete application of knowledge management: Lesson Learned Systems (LL-systems). These systems are mainly implemented in very large organizations where knowledge is often lost when an expert becomes unavailable. LL-systems are also used to prevent mishaps and failures by actively (or passively) warning the user of possible difficulties or problems. Examples for successful implementations of LL-systems are the Navy Lesson Learned System (NLLS) and Nasa's Lesson Learned Information System (LLIS).

Chapter 3 focusses on case-based reasoning, a method that is frequently employed in knowledge management. Lesson learned systems, for example, rely on this technique.

As an introduction, the case-based reasoning (CBR) approach, the nature of CBR-systems and different types such as interpretative CBR and problem solving CBR are discussed. The "Four Re's Cycle" describes the flow of information and the elementary processes in a basic CBR system. The four primary steps are: retrieve, reuse, revise, and retain.

Further sections about aspects of the implementation of CBR-systems and the use of CBR in educational environments demonstrate how CBR in general can be utilized in many KM applications. Finally, CBR is contrasted with some other, popular AI-techniques.

A KM system that handles only text-based information can be supplemented with CBR and similar functionality relatively easily. For systems that deal with multimedia information such as images, sound or videos, the approach to knowledge management is generally more difficult because the content cannot be accessed directly. In this case, the multimedia content has to be described using metadata before it can be processed by traditional techniques such as CBR.

Hence, chapter 4 introduces MPEG-7, one of the most recent developments in the description of metadata. Furthermore other metadata intiatives such as Dublin Core or the Open Archives Initiative (OAI) are explained.

The chapter gives an overview of the family of MPEG standards including MPEG-1, -2, and -4, as well as MPEG-7 and MPEG-21. After describing some details and core components of MPEG-7, two prototypes of MPEG-7 applications are presented: IMKA and TV-Trawler. IMKA is a system that allows the user to query for media objects on both the semantic and the perspective level.

TV-Trawler aims at digital video broadcasting. It automatically checks the content description of TV programs and records a program, if its description matches the user's preferences.

The last chapter of Part 1 deals with a topic that is related to metadata description: digital rights management. MPEG-21 and XrML as two major standards in this field are presented. The basic concepts of MPEG-21 are explained together with its digital item declaration, the digital item identification, and the rights expression language. The rights expression language of MPEG-21 is based on the XrML specification that is explained in detail.

Further sections of chapter 5 include related initiatives and competing standards such as the Internet Digital Rights Management (IDRM) Initiative, the Open Digital Rights Language (ODRL), the Digital Object Identifier (DOI) initiative, and the work of the Content ID Forum (cIDf).

Chapter 5 concludes the *Introduction to Knowledge Management,* Part 1 of this thesis. If readers are already familiar with the theoretical foundations outlined in Part 1, they can go directly to Part 2, *Extensions to Active Documents,* or to Part 3, *Active Digital Video Broadcasting.*

Data, Information, Knowledge

1

1.1 Introduction

In this section the term knowledge and its relation to information and data is considered. Then, the various differing kinds of knowledge-based systems are briefly introduced and the question is raised, how knowledge management is positioned within the field of knowledge-based systems. Furthermore, the meaning and actual purpose of knowledge management is discussed and two approaches are presented. Finally, the notion of a state-of-the-art knowledge management-system is described.

1.2 Data, Information, Knowledge

Many authors such as [Bellinger et al. 2000] understand data, information, and knowledge as different views of data. The term *data* usually defines raw data, something unrefined that has no significance, yet. A spreadsheet, for example, contains many values that are, by themselves, basically random numbers. The numbers per se do not tell what they stand for or what they are associated with. They are data.

Information, in turn, is made up of data. It is data with relevance and a determined purpose. In the example above one would realize that the numbers in the spreadsheet stand for the average temperature (given in degrees centigrade) of every day of a certain year.

Knowledge, on the other hand, is based on information. The generation of knowledge from information requires the process of learning. When the learner associates the new information with the already stored data and gives the information a meaning, new knowledge is produced. In the example given above, the temperature-information could be related to other information about climate and weather.

The application of knowledge and understanding can generate new information. In a process of learning and understanding, the fact that summer is the hottest time of the year can ultimately be derived from the temperature-information. Thus, new information and new knowledge is generated.

Wisdom is on yet a higher level than knowledge. It is a collection of myriads of knowledge-items that are aggregated with experience and age. The concept of wisdom requires reflecting upon knowledge and experiences.

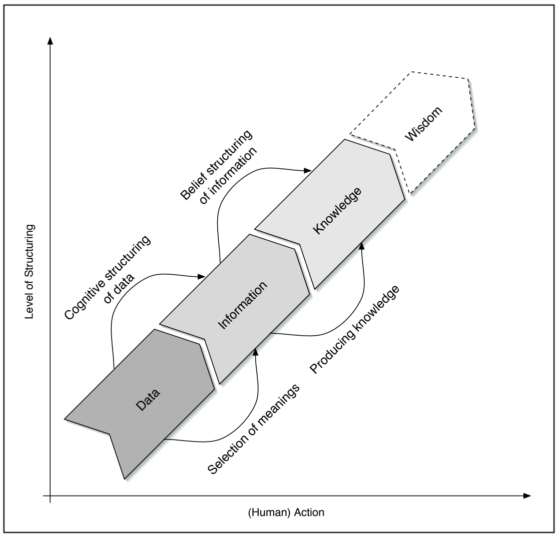


Figure 1: Data, information, and knowledge as different levels of structure, and the extent of (human) action required.

This definition of knowledge implies that one person's knowledge is another person's information! Everyone has to fulfill the acquisition-process themselves, knowledge cannot be passed on – only information can!

In an alternative model, data, information and knowledge are regarded as different levels of order or structural complexity. As graphically depicted in [Choo et al. 1998], data originates from a physical structuring of signals. Information stems from cognitive structuring of data, and knowledge results from so-called "belief structuring" of information. In this model, the order increases from signals to knowledge. Also, the amount of (human) action rises – from filtering on the signal-level to the selection of meanings on an information-level and actually producing knowledge. This model is shown in figure 1. (See also [Davenport 2002].)

1.3 Positioning Knowledge Management within Knowledge-Based Systems

Depending on size, functionality, and environment, three different branches can be distinguished in knowledge-based systems (KBS): knowledge engineering, knowledge processing, and knowledge management (KM).

Knowledge engineering focuses on technical issues such as the organization of knowledge-bases, finding optimal ways of representing knowledge, retrieving knowledge, etc. Its aim are ideal solutions to small-scale problems. It deals with micro-knowledge strategies, as it were (see [Tsui et al. 2000]). It could be compared to the design of databases or the development of optimal query strategies in information systems. However, its rather theoretical nature determines that knowledge engineering cannot be used unaccompanied in a system. Usually, it is applied "behind the scenes" and provides a framework to higher level applications. Again, there are similarities to database-applications: The database itself can never be used alone; it operates in the background and is controlled by higher level applications.

Knowledge management, on the other hand, does not deal with technical but with organizational, economic, and human issues. It tries to push sharing and reuse of knowledge in or among organizations and is concerned with macro-knowledge strategies. When KM does not include any technical aspects but only economic ones, it is referred to as "organizational memories".

The third discipline is knowledge processing. Sometimes considered a sub-topic of knowledge engineering, its main concerns are concluding new facts from existing ones and generating new knowledge. Generally speaking, knowledge processing is a practical but still very technical application of knowledge engineering that usually makes strong use of AI-techniques. Examples include expert systems, diagnosis, computer based design, and rule-based systems.

Although it would be desirable to merge all three technologies in order to combine and utilize the strengths of these research topics, they are often treated as disjunctive areas. Hardly any implemented KM-systems, for instance, use knowledge processing to generate new facts.

1.4 The Purpose of Knowledge Management

Although many authors claim that knowledge management helps delivering knowledge from a (computer) system to a human, in my opinion this notion bears a contradiction. As pointed out above, the acquisition of knowledge is based on an activity that has to be done by everyone, on their own. Thus, KM as such, cannot *directly* help in knowledge transfer. From that point of view, a KM-system can only facilitate and support the processes of gaining new knowledge. This could, for instance, be done by indicating cross-references the user would otherwise have not thought of or might not have been able to find. The system could also help the user find additional information on his or her request (e.g., [Alavi and Leidner 1999; Williams 2002]).

Another example is KM in big organizations (see [Maurer 2003] and [Baker 2002; Dieng-Kuntz and Matta 2002]). As can often be observed in multinational companies with several thousand employees, a developer in country A often does not know that there is an expert in this field in country B. A modern KM-system could keep track of the activities of all employees and try to resolve this redundancy or attempt to initiate mutual aid or a collaboration. Also, the distribution of information in general and the coordination involved is a great challenge. Clearly, this application is focussed on business and economic matters rather than technical ones.

For organizations such as the NASA Goddard Space Flight Center knowledge is a key asset. This, and the commitment to render NASA a knowledge-centered institution made it develop knowledge management- and knowledge sharing-initiatives ([Liebowitz 2002; Wu et al. 2002]). One project is the NASA Lessons Learned Information System (see section 2.5.2 and [LLIS 2002]). It is a web-based, automated database-application that collects experiences from over forty years of aeronautics and space flight. The aim is to prevent failures of the past and support information sharing, but also quality and safety can be incorporated at a very early stage. Lesson learned systems are characterized in more detail in chapter 2.

Yet another instance is NCR (see [Babilon 1998]). As many other large companies, NCR had to deal with an explosion of the information produced in the last decade. The need for a system that embraced all information-producing, -processing and -consuming areas of the company

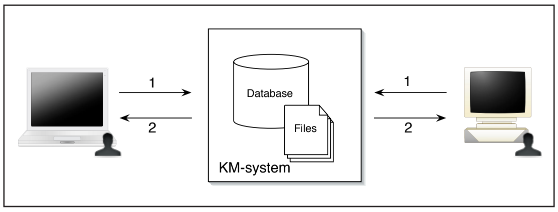


Figure 2: An elementary KM-system that bears large resemblances to classic information systems.

grew. What started off as an information management-system in the 1990s has now become a KM-system with sophisticated retrieval functions. NCR's KM-system was extended as a result of changing requirements and newly available technologies, which is a reason why it was introduced in several phases. However, its establishment did not only demand technical changes but also changes in the corporate culture.

One problem of the attempt to describe the purpose of KM is that such a system is usually not only focussed on one task but rather a broad variety of possible tasks. Thus, in my opinion, the main intent of KM is to facilitate knowledge transfer and acquisition of knowledge, and to offer new ways of retrieving and exploiting information that might otherwise have been unnoticed. Although KM should make use of the most current technologies available, its development is frequently driven by businesses and economic requisites.

1.5 Approaches to Knowledge Management

From a technical viewpoint, at least two approaches to knowledge management can be identified: one from the side of information management and another one from artificial intelligence.

On the one hand, KM-systems are evolving from simple data management systems and information or document management systems. Data management systems introduce little or no order in the data to be processed. An example is a file system: it imposes restrictions only on the filename but not on the contents or the type of files. On the next level are information management systems, where the demand for rules and a clear structure is growing. In a database, for instance, there can be rules for the type of data that is stored. Moreover, a clear organization (in tables) is employed. Finally, the structure of a KM-system requires even more sophistication. (This view is consistent with the explanation above and with figure 1.)

According to this, KM can be seen as emerging from information management, as an advanced document management system (e.g., [Borghoff and Pareschi 1997]). Actually, in [Allen 2002] "document storage and retrieval in a variety of formats" is mentioned as the simplest KM-application.

In contrast to that, KM is also entrenched in artificial intelligence (AI; see, for example, [Malhotra 2001]). From that point of view, KM is yet another application of AI. Whenever it is desired to not only store documents and information, but also to process the documents to obtain relationships among them or to harvest and produce additional information, AI might be used. Topics such as the automatic classification of documents, automatic generation of hierarchies, pattern and similarity recognition, and connection recognition rely on AI-technology and are successfully used in existing KM-systems. Bayesian networks and reasoning, ontologies, and intelligent agents are just a few more examples of possible uses of AI in KM-systems.

This thesis approaches knowledge management from an information systems-point of view and will attempt to make use of several AI-techniques.

1.6 Knowledge Management Systems

The most simplistic KM-system resembles a classic information management system. It contains either a centralized or a distributed storage system, which is usually a database, a method to put data into the repository (arrow 1) and one to retrieve it (arrow 2). Adding information to the system requires the user to explicitly call a corresponding function that might be called "add information or file". Retrieving information is usually based on a web-based query that is explicitly started by a user. The outline of such a system is depicted in figure 2.

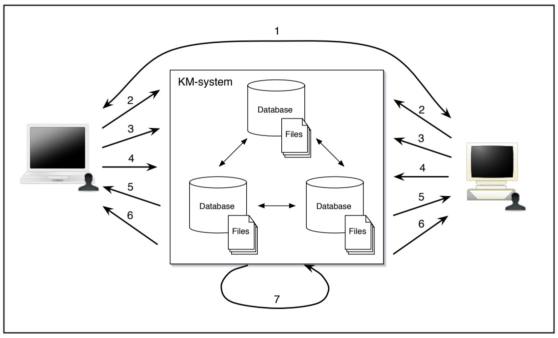


Figure 3: The Maurer-Tochermann-Model of knowledge management, redrawn from [Maurer and Tochtermann 2002].

A more sophisticated model was proposed as the Maurer-Tochtermann-Model (see [Maurer and Tochtermann 2002]), named after its authors. Its chief idea is to provide not only information flows explicitly initiated by the user, but also implicit input and output. The system is outlined in figure 3.

The illustration shows two users that are connected to the KM-system and the core of the system itself that consists in this diagram of several, distributed databases. The arrows constitute the possible flows of information in this system. In the following paragraphs the purpose of the different arrows will be discussed:

Arrow 1 stands for direct communication between the two users. This is "conventional" communication such as a phone-call or face-to-face conversation. Note, that the information that is exchanged via this channel is not captured by the system. However, considerable effort is being undertaken to capture spoken data in KM-systems (see [Moreno et al. 2002]).

The arrow labelled 2 describes the explicit information input that also exists in the elementary system in figure 2. Arrow 3 characterizes an implicit way of information entering the system. This information stems from the users' activities and documents. However, the user does not have to do anything in particular for the information to be absorbed by the system. This is a short-term process. The forth arrow specifies implicit information-input by analyzing the users' work on the computer. Many users have a distinct way of contact with the computer, complete tasks in a certain way, or use an individual schema to handle their software-applications. The data that can be collected from these observations might result in supporting the users in their particular ways of using the computer or helping them improve their experience in everyday work. In this respect, the forth arrow relates to a medium- to long-term strategy. Research areas such as case-based reasoning (see chapter 3) and gesture recognition deal with the underlying AI-techniques in more detail.

Arrow 5 indicates the explicit retrieval process that is also present in the simplistic approach (figure 2). Arrow 6 symbolizes that the system can offer knowledge without explicitly being asked by the user. This means that the system recognizes that the user might need help or that it finds resources in its database that is appropriate for the user's work. (See [Cyc 2002, OpenCyc 2002] for a comprehensive explanation of the Cyc-project on micro-worlds and contexts.)

Finally, arrow 7 explains that the system reflects upon the stored knowledge, that it processes the stored knowledge in order to generate new knowledge. This action could make use of knowledge processing, as described above.

The model describes all incoming and outgoing flows of information in a rather informal way. This design represents a framework that explains the basic idea and structure of a system (see also [Maurer 2003]). Nevertheless, it does not determine more – it does not say anything about how the information is put into the system, how it is processed, edited, and stored or distributed.

1.7 Conclusion

This chapter has clarified the differences between data, information, and knowledge and has pointed out the relationships amongst them at the same time. Based on that definition an attempt was made to describe the term knowledge management. Although several examples were provided in order to convey the idea of knowledge management, a single purpose of KM could not be identified.

This dilemma has, of course, also an impact on the definition of the term KM-system. On the one hand at least two approaches to a KM-system can be described, one originating in AI and the other one stemming from information management. On the other hand, however, it is hardly possible to describe the architecture of a KM-system. A few sophisticated and innovative models such as the Maurer-Tochtermann-Model have been proposed, but many authors still prefer to describe a KM-system in its most general form: at least one input and one output channel and a component for storing and processing data in between.

One thing is obvious, though: knowledge management will play an increasingly important role in future IT-systems, as large organizations willingly embrace the new technology.

Lesson Learned Systems

2

2.1 Introduction

This chapter presents the notion of Lesson Learned-systems, a way of reusing knowledge gained in the past. Among the organizations that use such systems are the Navy, US Air Force, the NSA, the Canadian Army, Nasa, ESA, the Japanese Space Agency NASDA, and companies such as XEROX and Microsoft.

After a description of the basic idea, the generic lesson learned process-cycle is presented. Then, problems with current implementations are examined and some examples are depicted. Furthermore, the relations to similar areas of interest such as case-based reasoning (CBR) or active documents are outlined.

2.2 The Basic Idea of Lesson Learned Systems

Large organizations, particularly in military- and government-environments, frequently experience the problem that valuable knowledge is lost when experts become unavailable and their knowledge should have been passed on to other employees. Thus, the need for an appropriate KM-module in the form of a lessons learned system (LL-system) arises. A prime incentive for companies to make use of LL-systems is that flops should be prevented and at the same time the quality should be constantly improved. These two demands incorporate learning from previous experiences in both technical and economic fields (e.g., [Liebowitz 2001; Liebowitz 2002]).

Defined in a more formal way, a "lesson learned is a knowledge or understanding gained by experience" (from [Weber et al. 2001]). These experiences can be positive or negative, can be successfully finished projects or mishaps and failures.

For lessons, the following conditions must be met: they must be valid, significant, and applicable. Lessons are valid, if technical aspects and the facts are correct, and other formal requirements are met. The significance is determined by the assumed effect it will have on a process, an operation, or a project. It does not express, though, if the lesson will ever be used. Finally, the applicability designates a certain task, process, or design that prevents or reduces failures respectively improves a positive outcome.

2.3 The Generic Lesson Learned Process

Typically, a lesson learned process consists of four activities: collecting lessons, verification, storage, and dissemination. The process starts with the collection of new lessons. Information can be collected passively, i.e. the user explicitly submits his data. Other ways of passive information-collection include mandatory completion of a form after a project has ended or dedicated interviews with users in order to gather information. Lessons can also be passively collected while solving a problem. But again, the system can only remind the user to more or less explicitly submit information on the status of a process.

In contrast to the passive accumulation of information, data can also be harvested actively. Scanning documents and communication among an organization is one approach to fully automated, active collection of lessons. As shown in figure 1, this first step always includes members from the organization.

After the lessons are collected in the LL-center, they have to be verified. This process usually involves human domain experts. However, advancing AI-technology might make it possible to introduce semi-automated verification-methods, in which the computer performs tasks such as classification of lessons or inspection of formal criteria.

Next, the verified lessons are stored in a repository that is represented by a relational database in most cases. Hierarchical storage methods such as filesystems or hierarchical databases are also employed at times.

In compliance with [Weber et al. 2001], I think that the dissemination of the lessons is the most important and precarious process within the cycle. As with the collection of lessons there are several ways to disseminate lessons – with not very astoundingly different results: Passive dissemination "waits" for the user to explicitly query the system for a specific lesson. Clearly, this is the least successful approach. The US-Navy, for example, uses such a system with more than 49,000 lessons. Despite the huge amount of information the system is hardly ever utilized.

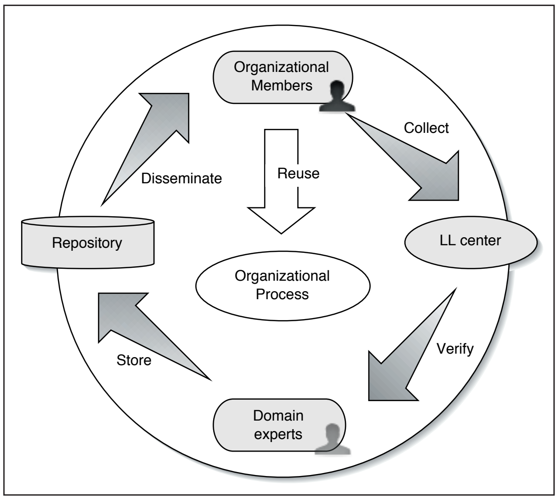


Figure 1: The generic lesson learned process-cycle.

In active dissemination the goal is to broadcast lessons to a potential list of uses. However, the user-list is usually managed manually, which poses certain inefficiencies: On the one hand, with a growing number of users, it is virtually impossible to put all concerned members on the list; on the other hand growing user-lists are very hard to overlook and manage. NASA with its lessons learned information system (LLIS, see [LLIS 2002]) apply a comparable approach in which user-profiles are exploited to distribute information only to a relevant group of employees.

Another noteworthy and very promising method is proactive dissemination. The actions and events in a user interface are stored on a per-user basis and are used to predict which lessons might be practical. Only these selected lessons are presented to the user automatically. A user, for instance, spends several days searching the organization's intranet and the internet for a specific, new programming language, but cannot find anything about it. Now, an expert adds a new lesson to the lessons learned system. By analyzing the user's "behavior", the system assumes that the lesson that was submitted by the expert is possibly useful, and informs the user about the new lesson.

The generic lessons learned process-cycle illustrates that after several steps the information that originated from a single person or a department can be reused throughout the whole organization and might be taken advantage of to improve the organizational process. The LL-cycle resembles the "four re-s"-cycle in case-based reasoning (see section 3.4). In fact, a lessons learned system can be seen as a CBR-system, as briefly discussed below.

2.4 Drawbacks of LL-Systems

Commonly, lessons are not available in a standardized format or structure. Some lessons are available as plain text-documents, many are formatted in HTML, some in SGML, and yet others are stored in proprietary formats. That causes trouble when trying to create a knowledge-base in a single, ordered structure. Many implementations do not solve the problem but only provide one monolithic, utterly unstructured field for the complete content of a lesson. As in most KM-applications this lack of a clear structure and strict organizational rules decreases the usefulness of the lessons learned-application, and the efficiency and quality of knowledge-retrieval. Retrenchments are made in favor of ease of implementation.

Clearly, it is rather easy to write loosely structured documents. On the other hand, that makes it hard for a computer-system to extract knowledge, which may result in poor performance-rates. That, in turn, could cause low acceptance and render a system futile. Therefore obligatory guidelines for writing lessons should be provided by an organization or company. Special tools for support in the authoring-process might be useful.

Another difficulty arises from the expectations of the economy in KM-systems: Companies demand immediate benefits such as increased profits and improved efficiency of processes. This is commonly achieved by collecting and storing all kinds of information and documents in a knowledge base. Consequently as many relations and links as possible are created among the information, and new knowledge is generated. Catchwords are "organizational memories" and "data-mining". What seems to be good for general-purpose KM-systems has negative effects on LL-systems: Although those so-called *hybrid repositories,* that hold various kinds of objects, increase the performance of other KM-systems, they reduce the effectiveness of LL-systems (see [Weber et al. 2001]).

In addition, so far no "generic" LL-systems are available. All systems are either task-specific or domain-specific or both. These domains, for instance, include planning or technical problems. Since the nature of these lessons is very different, an appropriate way of representation for either of them has to be found. Technical lessons are very often represented in the form of tuples such as <problem, solution> or <question, answer>. Planning lessons, on the other hand, have a significantly distinct structure consisting of an originating action, a set of conditions, contributions, and results, as well as suggestions and applicable tasks. It is very difficult to unify these two representations.

Yet another problem is that LL-systems are commonly only meant for support on a non-management level. Only very few exceptions, such as Nasa's LLIS, exist. Thus, the decision-making process does not benefit from the LL-facility. Furthermore this is inconsistent with the notion of a seamless integration of all flows of information in a KM-system.

2.5 Examples

In the next few paragraphs two systems, one maintained by the Navy and the other one by the Nasa, shall be presented. Both systems are employed in government organizations and pursue similar aims, yet the technical approaches are different.

2.5.1 The Navy Lessons Learned System

The Navy Lessons Learned System (NLLS) was developed to meet the explicit demand for a easily accessible, central lessons learned database. It should provide worldwide access to documents such as information lessons, summary reports, and after-action reports ([NNLS 2002]). The first version of the NLLS was released in 1991 and was designed around existing software products, in order to keep costs as low as possible. It was a standalone system with a passive, browsable lesson dissemination process. This approach was prone to fail because it was based on unrealistic user assumptions. The problems are that

- users are very often not familiar with NLLS or do not have access to it;
- users do not always have time and the skills to search for lessons;
- users can sometimes not correctly use, digest, and reuse lessons; and
- users are not reminded of lessons, when they actually need them.

In a new proposal two tools, ALDS and LET, assist the collection- and dissemination-processes ([Aha 2002b]). ALDS is a proactive lesson distribution system that assesses the relevances of lessons for upcoming problems. Appropriate lessons are retrieved and recommended to the user. LET, the second utility, supports lesson-authors by guiding them through the process of creating new lessons. Moreover, for the extraction and discovery of knowledge from existing, unstructured lessons, a semi-automated procedure can be applied.

2.5.2 The Nasa Lessons Learned Information System

Nasa Goddard Space Flight Center operates an LL-system called LLIS (sometimes also named RECALL). Nasa's intent to employ such a system was especially the fact that in mission critical situations mistakes recurred and led to failures. The system operates on a per-user basis. Whenever a new user wants to join the LL-system, a user-profile has to be completed. Consequently, in a process named *active casting* all users with matching profiles receive only the content that is relevant for them.

LLIS uses text and <question, answer>-pairs to represent lessons as cases. That is necessary because the system includes the case-based reasoning methodology to improve retrieval. One benefit of this method is that the user can search in a so-called constrained dialogue. A drawback is, though, that a significantly higher amount of knowledge engineering must be dedicated to designing and implementing such a system.

Nasa's LL-system is not only available to researchers, engineers, and technicians, but also to project managers, for instance. According to [Lee 2002], the system is used by project managers and general management for process- and project-risk management. In doing so, the probable risk of a current project can be traced back to failures or successes of previous ones.

2.6 LL-Systems and Case-Based Reasoning

Case-Based Reasoning (CBR) is described in more detail in chapter 3. An intent to use CBR is reuse of information and all kinds of experiences within an organization to solve problems in similar ways previous problems were solved.

Both CBR-systems and LL-systems try to find experiences from the past, analyze them, process them and store them in a (public) repository. These resemblances can clearly be seen in the LL-cycle and the CBR-cycle, respectively.

In a way, LL-systems make use of the same idea, but restrict the range of experiences involved. Only experiences (lessons) that are capable of being used in the future will be examined, processed and stored. In contrast to that, CBR in its loosest definition makes use of as much information as possible. The requirements for a lesson to be valid, significant and applicable do in theory not hold for a general case in a CBR-system.

The biggest difference is the role of the human factor in the two technologies. In LL-systems people are very important. Employees and experts are used to collect information, assemble the lessons, to assess and verify them, and to finally use them again after their dissemination. In an ideal CBR-system no people should be involved in the (technical) process as such. From the collection of data to its verification and storage, no human beings are needed. Not even the consumer of the output of a CBR-system has to be human!

Finally, the scale of the systems is also divergent. LL-systems are usually big, standalone systems the user interacts with directly. A typical LL-application could be compared to a database-application with a web-interface, where the user interacts with the web-interface. CBR-systems are very often only part of a yet bigger system. In that, CBR-systems are in the background and do normally not offer interfaces to the end-user. An ordinary user does usually not interact with a CBR-system directly.

2.7 LL-Systems and Active Documents

Both LL-systems and active documents pursue the mutual aim of proactivity – in different respects, though. Active document-facilities embrace all kinds of documents and any type of content, without restrictions. As illustrated in chapter 8, the idea of active documents is applicable to almost any kind of document, and a certain structure or special preparation of documents is not entailed.

LL-systems, on the other hand, depend on a structure and are confined to a particular task or purpose. Also, checking the prerequisites for a lesson can result in rather high expenses, whereas active documents usually do not cause much effort. While for active documents the primary focus is on interaction with documents, LL-systems are clearly determined to push reuse of information and knowledge.

Lessons can be understood as a special sort of document. As such, a lesson can incorporate all features of an active document, can be an active document. Active documents, in turn, could also include features from LL-systems, such as the dissemination-process. Techniques from active and pro-active dissemination of lessons could be applied to the propagation of annotations to documents that were recently processed by the user (see section 7.3).

The observation from a higher level shows that active document-systems are add-ons to existing document- or information-management systems, whereas LL-systems are very often standalone-systems. Active documents provide a certain feature, whereas LL-systems provide a whole new application.

2.8 Conclusion

Lesson learned systems are an example for a not extremely complex but yet quite successful KM-application. With their help knowledge can be passed on more easily, possible problems and failures can be identified at an early stage and prevented.

However, LL-systems also clearly show the problems many other KM-systems also share: the source material usually has no structure and no common format, the integration in existing systems can be very difficult, lessons have to be authored according to strict guidelines, and the presentation of information together with the information dissemination process is a crucial factor.

Case-Based Reasoning

3

3.1 Introduction

Case-based reasoning as an innovative method for problem solving is frequently employed in knowledge management systems. This chapter gives an overview of the basic ideas behind case-based reasoning (CBR). First, the CBR-approach and how it conforms with the human way of reasoning is presented. Then, the differences between interpretative and problem-solving CBR and their according uses are detailed. The traditional sub-processes of CBR are demonstrated using the four "re's"-cycle. Moreover, some aspects of the implementation of CBR-systems, and the role of CBR in educational environments are described. Finally, the relations of case-based reasoning to other AI-techniques are discussed.

Throughout this chapter examples for applications of CBR are provided.

3.2 The CBR-Approach

3.2.1 What is CBR?

CBR is a general technique for solving problems. Given a problem, a suitable solution has to be discovered. In the traditional reasoning cycle in cognitive psychology a reasoner analyzes a problem and can solve it by composing and instantiating abstract operators (e.g., [Herzog 1994]). Consequently, the same problem or partial problems are solved every time they appear. This technique is often employed in artificial intelligence and rule-based systems. Compared to that, CBR makes use of a different methodology.

The basic idea for CBR comes from the way humans remember things: When a problem has to be solved, humans do not start from scratch, but rather remember how they solved the same or similar problems in the past. That concept implies that the solution for the given problem might resemble the solution of another problem that was encountered in the past. In other words, CBR is based on the assumption that the world is regular.

A benefit of CBR is that it is generally a fairly robust technique: Humans, for instance, solve difficult problems based on uncertain and sometimes limited knowledge, and additionally, the performance increases with experience. The same can basically be true for machine-based CBR.

3.2.2 Achieving CBR

In a CBR-approach, whenever a problem has to be solved, a case-based reasoner first searches for similar and already solved problems. Then, the closest matching case is located. The task of selecting an adequate case from the so-called *case-base* is one of the most important ones in CBR. This is the *indexing problem*. After inspecting the actual differences between the (already solved) case and the problem, in a process named *inference* the experiences from the known case are applied to the new one. In doing that, this technique resembles the human way of solving problems (e.g., [Aha 2002a; AI-CBR 2002; Aitken 2001]).

An example for CBR in everyday-life can be found in law and in the courts: in Anglo-American countries, the law-system is based on common law, as opposed to Roman law in Central Europe, for instance. When a decision is to be made in court, the current case is compared to existing findings (precedents). In general, similar cases will result in similar court decisions in this system.

3.2.3 CBR-Systems

A CBR-system typically consists of a knowledge-base or experience-base that stores all known problems, so-called cases, and experiences connected with them. Experiences can be concrete solutions to a problem, intermediate results, even by-products or side-effects, and methods that were used for obtaining solutions, references to other cases, etc. Not only successes may be retained but also failures that can be used to warn the user of potential pitfalls and discrepancies. What experiences are actually stored in a particular system may vary and depends strongly on the application. Usually, it is not reasonable to accumulate all information that is available, though.

The description of the technique in the paragraphs above includes a broad variety of possible systems and implementations. However, most existing CBR-systems are based on a narrower view and a stricter definition, which results in less complexity and easier programmability.

3.3 Interpretative CBR and Problem-Solving CBR

CBR can be divided into interpretive CBR and problem-solving CBR. Both methods are used in practice and have numerous successful applications.

3.3.1 Interpretative CBR

The aim of interpretative CBR is usually to classify a new case, to „interpret“ which group of cases it belongs to. Hence, interpretative CBR is an analytic method.

A category is found by comparing previously determined features of the new case to those of already categorized cases. Traditionally the classification process involves four steps (see figure 1): First, the new case has to be assessed to find out which features are relevant. Then, the relevant features are used to query the case-base and to find similar cases. In a third step the retrieved cases are compared to the current problem in order to find an appropriate problem-class. Finally, the new case together with the classification-information is stored in the case-base.

A successful application of interpretative CBR are systems carrying out medical diagnoses of cancer, where the type of tissue is determined from the observation of several attributes. These attributes form a new case that can be compared to previous cases. This classification ultimately yields a suggestion whether cancer treatment is necessary or not. Note that diagnosis is not only limited to the medical domain but also includes the analysis of (faulty) technical systems.

3.3.2 Case-Based Problem Solving

Case-based problem solving is used to generate a solution for a new problem from solutions of already solved problems. From that point of view, case-based problem solving can be understood as a synthetic task.

The four step-process of interpretative CBR illustrated in figure 1 has to be extended by another component that analyzes similarities and differences between the cases in order to find out which parts have to be adapted to meet all conditions of the new situation.

This branch of CBR is utilized in fields such as case-based design, planning, and explanation systems. An example is planning a transportation system: Although two transportation systems will never be exactly identical, prior implementations can be largely reused (e.g., [Khattak and Kanafani 1996]). Therefore an new transportation system is usually based on an existing system. Imagine that the Vienna city council decides to implement a new transportation system. In a first step, a model with similar requisitions has to be found. Munich could be retrieved as the most appropriate basis for the further process. Then, dissimilarities such as deviating laws, availability of resources or a different geology must be identified. In order to meet all constraints, the encountered case (transportation system of Munich) has to be adapted to solve the new problem (Vienna).

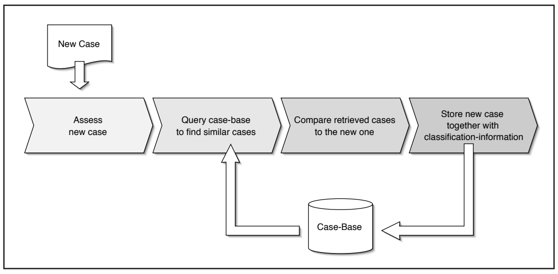


Figure 1: Operations involved in interpretative CBR.

In suchlike systems the CBR-component finds and proposes existing systems that closely match the current problem. The adjustment of the encountered case is usually not carried out by the computer-system. It only assists in warning the user of complications and possible mishaps.

3.4 The Four “Re’s” of CBR

The four „re’s“ of CBR play an important role (e.g., [Watson 1997], [AAAI 2002]) in that they are forming a cycle that describes the basic order of events in a CBR-system. They are: retrieve, reuse, revise, and retain. These sub-processes can be graphically depicted in a general CBR-cycle as shown in figure 2.

3.4.1 Retrieval

The cycle starts with retrieving the most similar case for a new problem. Retrieval, and the closely linked operation of storing the cases, often determines the quality and usefulness of a CBR-system. Therefore many different ideas for retrieval (and storage) have been proposed, and sometimes even non-textural information such as images are considered.

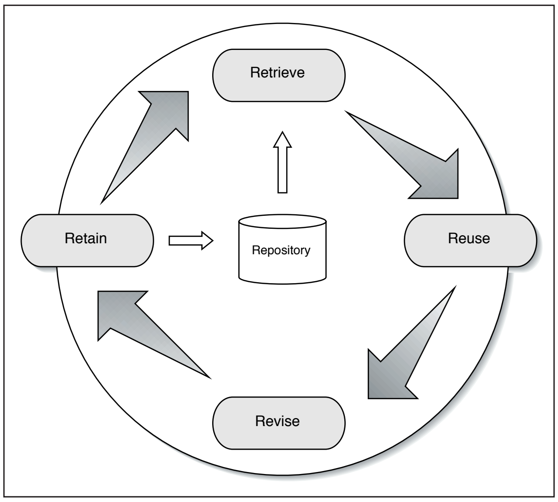


Figure 2: The four "Re's" of CBR.

There is no such thing a one single "best" retrieval-process because it depends on both the nature of the application and very often also on the domain in which it is used. In educational CBR-systems, for instance, cases are retrieved, but usually adaptation is not employed because this is left to the learner. On the other hand, in planning systems adaptation is appreciated and necessary, which raises the question which cases are generally advantageous: cases that closely resemble the new problem, cases that can easily be adapted, cases that have already been successfully adapted, etc. On top of all, retrieved cases have to be *usefully* similar.

Compared to common information retrieval and database systems, retrieval in CBR tends to be more active. A database query has to be as precise as possible because it is based on exact (text) pattern matching. In contrast to that, queries in CBR-systems need not be exact, they can even be carried out if they are partly incorrect or knowledge is incomplete. The system itself will compensate for these shortcomings and will possibly not present exact matches but rather select the most similar cases.

3.4.2 Reuse

In the second step of the cycle, the experience and knowledge that is associated with the case is reused to solve a new task. This process, inference, commonly refers to the adaptation of cases. Historically speaking, case-adaptation stems from analogical reasoning.

Adaptation includes operations such as generalization, specialization, and substitution. Generalization, for example, is needed to apply the knowledge of a single case to a whole class of cases. Specialization defines virtually a new sub-class based on a single case. This leads to a hierarchical view of the case-base.

The adaptation sub-process is very often done by rule-based systems (RBSs). Since RBSs do not operate in a satisfactory way without it, a domain theory is required. Consequently, the use of the CBR-system will be restricted to a certain domain because in other domains the case-adaptation would cease to work.

The RBS needs both a domain theory and a formulation of the theory of case-adaptation to be capable of manipulating cases. In an attempt to circumvent static definitions, also the case-adaptation sub-system can be seen as a CBR-system. Thus, the adaptation-process should learn from previous successful (and failed) adaptations.

A different way of making case-adaptation easier is to revise and alter the task itself in order to reduce the necessity of adaptation as such. Both RBSs and CBR-systems could be employed to achieve this. Again, with the involvement of a rule-based component the side-effects of comprising domain knowledge have to be accepted. The use of a CBR-system, on the other hand, leads to a recursive definition of the problem.

As already mentioned above, some systems completely avoid the adaptation problem. While the common CBR-cycle suggests the automated adaptation of retrieved cases, another approach is to only propose these cases. The adaptation as well as the revision are then carried out by humans. These systems are sometimes called *retrieve and propose systems*. Helpdesk systems, for example, make use of the retrieve and propose paradigm. When a person working in a support center has to process a customer query, a request is issued to the CBR system to find out if similar problems have already occurred. The reply of the system provides matching cases but does not include any adaptations.

The general class of this type of systems is known as *case-based aiding systems*. What they all have in common is the provision of a case-memory whose task it is to assist humans in making decisions by providing prior successful examples as well as avoiding previous mistakes. Still, the human reasoner retains the full control over the decision making process – the system only informs and supports the user. Studies show that users of such systems readily accept them and appreciate their availability.

3.4.3 Revise

After the case-adaptation process the proposed solution has to be revised. This process is commonly referred to as *similarity assessment*. Similarity is determined by comparing certain features of old cases and the current task to the adapted case. That poses the problem of finding out if appropriate features are compared to do sensible similarity assessment. A need for robust similarity metrics arises. Practical examples show that suitable features can be determined more easily when a good and strict description of the domain knowledge is available.

In constructive similarity assessment, which is one applicable algorithm for that task, similarity is provided if the input problem can be transformed (constructed) to the selected, adapted case in a finite number of steps.

3.4.4 Retain

If it is a valid solution to the initial problem, the new experiences constitute valuable information for future requests to the system and should be retained in the case-base.

Amongst other advantages, a benefit of retaining cases is that decisions are easily explainable to users. Furthermore, they can be verified by the user.

In order to provide more flexibility, and speedup the whole CBR-process, not all cases should be stored. In fact, the case-base should only be extended if (and only if!) the initial case-base is insufficient for a task. Precise cases that are neither too specific nor too general should be preferred instead of loosely defined ones. This notion helps avoiding contradictions and ambiguities in the case-base.

The size of a case-base cannot be stated in a general way since it strongly depends on the kind of application. Some tasks might require only very few cases to work satisfyingly, whereas other applications can demand several thousand cases (see [Leake 1996]).

3.5 Implementational Aspects of CBR

3.5.1 Case-Bases

Cases can be stored in several ways. The straightforward approach is to store each case as a monolithic data-element.

Attribute-value vectors, on the other hand, offer a slightly different way of representing cases. An attribute-value-pair is a tuple that describes a certain feature of a case. Thus, attribute-value vectors are feature vectors. Although feature vectors introduce a minimum of order and structure, they are still seen as a flat representation.

These „flat“ approaches frequently limit the capability of systems to adapt cases. Hence, another idea is to break down the cases into sub-cases – so-called *snippets*. These snippets could be stored in an object-oriented or hierarchical manner describing sub- and super-principles ([Lenz 1998]). A different, promising approach is to retain cases as traces of how a solution was reached. This enables the reuse of paradigms in problem solving and makes the reuse of only a particular section of a case easier.

Interpretative CBR uses *decision trees* to expedite classification. Inner nodes of the tree are attributes, the connected edges are the corresponding values, and leaves stand for classes. A path in the decision tree states that a case with a certain feature vector belongs to a particular class.

3.5.2 Learning in CBR

CBR systems implement incremental learning, i.e. the correctness of the performance does not depend on the number of cases stored in the case-base. If a case can be related to a suitable case from the memory this relation is correct. So even if there are only few cases available, all conclusions related to these cases are correct. However, the system might not be able to process all queries due to the lack of appropriate cases.

CBR is a *lazy learning technique* in that it does not process cases in its case-base after they have been stored. No offline learning takes place. Only on an explicit request by a user are cases retrieved and adapted. Thus, learning happens only when a user interacts with the system. Therefore the quality of the system and its successful learning depends to a certain extent on the utilization of the system.

3.5.3 Separation of Problems and Solutions

In most systems a separation of problem (commonly referred to as case), solution (sometimes called case-completion), and inference component is suggested. Cases can be seen as relations between problems and solutions, as <problem, solution>-tuples. However, this view does not grasp the whole scope of cases and their possible consequences. Therefore the trivial model is extended and finally uses <problem, solution, (side-)effect>-tuples.

3.6 CBR in Educational Environments

In a CBR context, the generic term for educational and learning environments is knowledge-based tutoring and help-systems, and their intent is to support users (learners) individually. CBR is often a core component of of these systems. This section describes the differences between static and dynamic learning systems and explains how CBR can be employed.

3.6.1 Static and Dynamic Learning Systems

As detailed [Weber and Schult 1998] conventional, static case-based learning systems only use pre-defined cases that are compiled by an instructor. These cases are utilized when new topics are to be explained. The advantage is that these examples can be carefully selected and prepared by the instructor, and their incidence can be well chosen so that they best suit the intended purpose. A disadvantage is that these cases are not responsive to the users' needs and their individual way of learning. This problem is similar to a lecturer‘s dilemma: Although a lecturer wants to provide a "perfect lecture" for all persons in the audience, it might be virtually impossible to do so because of the individuality of every auditor. Some students, for example, might prefer a flaring up speech, while others incline towards an even-tempered talk; some people like animated, rich, and colorful presentations more than black and white slides; etc.

Dynamic systems, on the other hand, are adaptable and adaptive. The system collects new cases by observing the users and analyzing their problem solving methods. When similar situations occur, i.e. the user has to solve conceptually or contextually similar problems, the system is able to actively support the user. That way the system could also find out when the user needs an example, or it could show references to related areas when the user is stuck in solving a problem. Dynamic systems are particularly advantageous for users that are novices in a certain domain. For them generic examples occasionally can be too difficult to understand or they are not able to connect the examples to the theoretical foundations they learned ([Weber and Schult 1998]).

In general, the notion of providing individual support for users can only be realized using dynamic learning systems.

3.6.2 The Use of CBR

Learning environments can make use of both-problem solving CBR and interpretative CBR (as described above).

Interpretative CBR, for instance, can be used to adapt to the user. When the user makes the same kind of mistake several times, a more thorough explanation of the underlying theory can be provided. The system classifies all mistakes, and when the number of mistakes of a particular class a person makes exceeds a threshold value, an explanation component is instantiated. The collected information can also be exploited to provide preventative measures: The user may as well make the same mistakes as several learners have made before. The system recognizes the potential pitfall and tells the user to be exceedingly attentive. This notion aims at predicting the user's way of solving problems or completing tasks (e.g., [Weber and Schult 1998]).

Interpretative CBR is usually combined with a static learning system-approach because the concepts match well. Furthermore the classification process is less complex. Loosely defined domains and complex problems are not well manageable in these systems, though.

In web-based training (WBT) systems, for example, the user is frequently expected to complete some examples at the end of each chapter. This is where problem-solving CBR sets in: The given answers (new case) are compared to pre-defined, correct answers (old, retained case). If the two cases are identical or sufficiently similar, the answer is considered to be correct. Otherwise (the answer is incorrect or incomplete) the diagnostic component of the CBR system analyzes the provided solution in order to find out if the solution path would have been correct. In the latter case the system can point out why the provided answer was incorrect.

Even more than in common purpose CBR-systems, the amount of the domain knowledge is essential. The narrower the domain is, and the exacter and more specific the cases are, the better the system can respond to requests. Also, the task or assignment has to be described precisely and in a structured way. However, the cost of greater domain knowledge can increase the performance and usefulness of learning systems significantly.

For further examples and advanced applications of CBR, see, for example, [Althoff et al. 1998] or [Burkhard 1998].

3.7 CBR and Other AI-Techniques

3.7.1 Differences Compared with Other AI-Techniques

CBR-systems perform tasks similar to other AI-techniques such as neural networks or rule-based systems. Still, they offer some advantages that other architectures lack.

While other architectures such as generative AI-systems must deal with all possible cases of a problem-space, CBR-systems only need to cope with problems that actually can occur in practice. Usually, that greatly reduces problem-space and makes the management of the knowledge-base less complicated.

Neural networks are often used for pattern matching and recognition or for classification tasks. As pointed out above, classification is also one of the most important applications of CBR. The approach to achieving this goal is completely different, though. An artificial neural network (ANN) is a set of information processing elements that are highly interconnected. The learning process that consists of training iteratively customizes the weights of the connections (synapses). These connection weights maintain all the information that is necessary to solve problems. However, this information is not sufficient to explain how the system reached a conclusion, which is a big disadvantage compared to CBR.

Rule-based systems (RBS) are often compared and also contrasted with CBR (e.g., [Aamodt and Plaza 1994]). A certain set of rules that is stored in a rule-base is applied to either old or new facts in order to generate new knowledge. A benefit of RBSs is that the system itself can explain how it reached a conclusion. Although there are many very successful examples of RBSs, they have some drawbacks such as difficulties in managing complex rule-bases with several hundred or thousand rules.

One of the most popular examples for a RBS is Mycin, an application of RBSs in a medical environment (see [Buchanan and Shortliffe 1984]). The system uses very simple rules that are similar to traditional if-then-statements. A very simple rule could be "if fever(patient) then sick(patient)", to denote that a patient is sick, if high temperature can be diagnosed. The user (patient or doctor) enters facts about a supposed disease such as rash, high temperature or pain. After having applied rules of the rule-base to these facts, the system reaches the conclusion that the patient suffers a particular disease. Furthermore, the system tells the user to which percentage this diagnosis is true. Note that the RBS only uses the submitted facts and the pre-defined rules to come to this conclusion – previous experiences (diagnoses) are not employed.

3.7.2 Integration of CBR and Other AI-Techniques

Although CBR can be used as the only AI-technique in an application this is rather the exception from the rule. Other concepts and methodologies are commonly used to support case-acquisition and -adaptation or to refine the query-process.

Rule-based systems, for instance, are clearly discriminated from CBR. Nevertheless, CBR-systems are often augmented by rule-based elements. When no suitable case can be retrieved or the case-adaptation fails, certain rules in a rule-based backup system can be applied in order to derive the desired facts.

On the other hand, CBR can be employed very well by many other AI-techniques: Whenever it is expensive to start deductions or reasoning from scratch, CBR may be used: Before instantiating the actual AI-process, a lookup in a case-base may show that the same problem has already been solved.

If the case-base is organized in a particular way, only parts of stored cases (snippets) can be reused. Planning a dinner for five people, for example, could be an application: The host remembers that he successfully cooked dinner for six people. This time, the number of guests is smaller, and one person is vegetarian. Thus, the host can use the same recipe as last time (lookup in the case-base) but has to make some adaptations, such as cooking for only five people and providing a vegetarian meal (application of AI-methods).

Hence, CBR can help to reduce the computational costs in other AI-technologies by providing a facility to access experiences and previously processed examples.

3.7.3 Acceptance of AI-Techniques in General and of CBR in Particular

AI-techniques are often not well accepted by users. This is due to the fact that the user can often not see or reconstruct how the computer derived a fact or came to a conclusion. With artificial neuronal networks, for example, it is impossible to explain how a decision was reached (e.g., [Keller 2001]). Rule-based systems offer substantial improvements because the decision-making process can be described by the rules and the particular order in which they were applied to an initial fact or condition. However, rules are often too abstract or too hard to understand for users; sometimes users just do not accept rules.

Compared to that, CBR-systems are based on actual cases that can easily be exerted to give the user explanations for a better understanding of how certain decisions were made or new facts were generated.

3.8 Conclusion

This chapter presented case-based reasoning (CBR) as a flexible and efficient problem solving method. The basic paradigm was described together with the two approaches to CBR: problem-solving and interpretative CBR. Several examples have shown that the concept can be applied in as diverse areas as planning, design or explanation systems, and even e-learning environments.

In this thesis, chapter 7, *Extensions to Active Documents,* makes use of CBR functionality. Although not explicitly mentioned, also several other technologies introduced throughout later chapters can employ CBR to improve the performance and the quality of services.

MPEG-7

4

4.1 Introduction

This chapter introduces MPEG-7, a relatively new standard in the MPEG-family. Unlike prior MPEG-standards it does not focus on the compression of audio-visual content but the description of multimedia data to pave the way for advanced applications such as content-based retrieval.

The chapter starts with an overview of the MPEG-standards, and the positioning of MPEG-7 within this series. MPEG-7 and its core components are described in greater detail, and two promising applications implementing the new standard are presented. Several other metadata initiatives such as OAI and Dublin Core and competing approaches to content-based retrieval shall be pointed out.

4.2 The Family of MPEG Standards

MPEG stands for Motion Picture Experts Group, a permanent working group of the International Standards Organization/International Electrotechnical Committee (ISO/IEC). It is in charge of developing standards for representing and coding digital audio and video information. When the MPEG was founded in 1988, the focus was on bit-efficient representation of audio-visual content, i.e. compressing, transporting, and decompressing audio-visual information. Best known are certainly the MPEG algorithms for compressing video and audio data.

However, more recent work focusses on the use of metadata, content-based retrieval and digital rights management. Therefore the chief purpose could be described as an all-embracing set of rules for the production, transmission, and management of digital media.

The target group MPEG is aiming at are home users (home entertainment and consumer electronics), PC and internet users (CDs, streaming video, web-radio, etc.) rather than studios (broadcasting and film industry) or other high bandwidth/bitrate applications (see [MPEG-2 2000]).

So far, MPEG-1, MPEG-2, and MPEG-4, as well as MPEG-7 and MPEG-21 are available. Their key features are briefly presented in the next few paragraphs.

4.2.1 MPEG-1

MPEG-1, approved in 1993 and officially named ISO/IEC 11172, is the standard for "coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s" ([MPEG-1 1996]). The standard includes five parts, of which the first three are most important and shall be shortly outlined.

Audio and video signals are processed separately. From the video signal single frames are selected periodically. The compression algorithm tries to causally predict the pictures between two of these base-frames using methods such as motion compensation. When the video signal is to be stored, only the base-frames and the motion compensation information are retained. Note that this compression technique is designed for rectangular video data such as TV-programmes. This algorithm for compressing video signals is described in Part 2 of the MPEG-1 standard.

Part 3 defines the coding of audio data, both mono and stereo. Basically, the audio compression algorithm omits tones that cannot be heard by humans (too high, not loud enough, etc.). The popular MP3-format is based on this technology, and in fact, its official name is "MPEG-1 Audio Layer 3".

The coded audio-visual data is transmitted in one single stream that includes the merged audio and video signals. The first part of the MPEG-1 standard defines how streams are assembled and combined for digital storage or transmission.

Examples for MPEG-1 applications are the VideoCD, MP3-players or computer programs for generating and playing MPEG-encoded movies.

4.2.2 MPEG-2

MPEG-2 (ISO/IEC 13818, [MPEG-2 2000] and [Chiariglione 2002]) was finally approved in November 1994 and is very similar to MPEG-1. It is built on the video compression technique of MPEG-1, and additionally allows the use of several streams for audio, video, and user data. The MPEG-2 audio coding scheme provides multichannel capabilities and is backward compatible to MPEG-1.

The standard supports the transition from analogue to digital satellite and cable transmission. It is used for digital television and, most important, for Video-DVDs. An overview of MPEG-2 in digital television is presented in section 10.2.1.

A historical note: MPEG-3 was designed as a standard that should comprise HDTV (High Definition TV). Since there were too many similarities, it was included in MPEG-2 in 1992.

4.2.3 MPEG-4

The requests of authors and service providers were addressed in MPEG-4 (ISO/IEC 14496). One of the key features is that the standard eliminates the limitation of a rectangular area by introducing *regions* (objects) that can be of virtually any shape. These objects can be of any type of media and are usually assembled in a scene.

News on television are an example. The scene consists of three elements: the background (with a picture of the current topic, for instance), the TV-presenter, and the voice of this person. The voice and the motion of the presenter have to be synchronized, while the picture of the presenter is positioned on the background-image. In other words, a scene in MPEG-4 creates a relation in time or space or both among objects.

MPEG-4 is the first of the MPEG-standards to offer interactivity to the user. Depending on the features included by the author, users may do things such as changing the viewing point of a scene, moving objects within a scene or triggering certain events ([MPEG-4 2002]).

Moreover, MPEG-4 takes important topics such as Quality of Service (QoS) and Intellectual Property Management and Protection (IPMP) into account. Hence, it presents significant improvements compared to MPEG-1 and MPEG-2.

MPEG-4 is considered to be the upcoming standard for interactive television, the internet and mobile computing (e.g., [Haas and Mayer 2001]). However, a restrictive licensing policy delayed the broad launch of the new technology. Recently Apple, as a leading multimedia software producer, included MPEG-4 into version 6 of its Quicktime software package ([Apple 2002; Quicktime 2002]).

4.2.4 MPEG-7

The previous standards, MPEG-1, -2, and -4, specify how to *represent* content. MPEG-7 (ISO/IEC 15938, see [MPEG-7 2001; MPEG-7 2002]), on the other hand, defines how to *describe* content. The central application of the MPEG-7 standard is content-based retrieval.

Content-based retrieval means retrieval based on the actual content of a document, not the filename or the location. This can be straightforward for text documents, but is in general rather difficult and a challenging task for other kinds of media such as images or videos. The state-of-the-art approach is attaching information that characterizes the content to the document itself. This information is commonly referred to as metadata.

MPEG-7 defines a framework for describing the content of any kind of data including text, images, motion pictures, and audio. These descriptions range from "simple" metadata such as the name of the author or the date of when a photograph was taken, to low-level features of objects, structural information, and even semantic relations (e.g., [Day 2001]).

The MPEG-7 standard is independent of the document-type and -format, codecs (for audio-visual information) or the storage-format and medium. It can be utilized with an MPEG-1 coded movie or with an MPEG-4 object, with a JPEG compressed picture or an AIFF coded sound-file. The standard does not define how the descriptions are to be generated or how they might be used.

Details and the core components of MPEG-7 are described in section 4.3 below.

4.2.5 MPEG-21

MPEG-21, which is not available as a standard, yet, addresses digital rights management. It is designed to provide commerce in digital content (publishing of digital movies and music, electronic books and images, etc.), while it takes copyrights and property management into account.

Digital rights management and the MPEG-21 draft ([MPEG-21 2002]) are discussed in greater detail in chapter 5.

4.2.6 The Big Picture and Future Expectations

While MPEG-1 and MPEG-2 have been used for several years and applications such as the VideoCD and DVD have already proved their strengths, the other three MPEG-standards still have to show their usefulness and effectiveness. A few MPEG-7 systems, for instance, have been proposed and some prototypes (see below) exist, but whether the standard will be a success, remains to be seen. MPEG-21 has not even been fully standardized, however there are already several competing initiatives (see chapter 5).

The expectations are great, and solutions to issues including ubiquitous access to media objects (mobile computing), digital rights management, better retrieval technologies for multimedia documents are anticipated. Some publications such as [Haas and Mayer 2001] or [Koenen 2001a,b] propose combinations of MPEG-4, -7, and -21.

MPEG-4 as a method for the representation of the content is particularly desirable because it allows for more detailed and sophisticated descriptions of the content. In a representation using MPEG-1 or MPEG-2, the complete broadcast is one made up of one or more streams, and no separate "objects" can be identified. Thus, a content-description and the specification of the corresponding permissions can only be applied to the complete movie (respectively the separate audio- and video-signals). When MPEG-4 is used to represent this scene, it can be described using several objects. The following example of news on television (see above) could define a new kind of interactive multimedia application. The TV-presenter is one object, whose MPEG-7 description could contain the person's name and a hyperlink to the corporate homepage. Music in the background is another object. The MPEG-7 description could contain the title of the song, the artist's name, the producer, a hyperlink to the artist's website, and other details. The MPEG-21 data adds information about the digital rights, if the video track may be extracted, and under which circumstances the audio track may be reproduced or reused, etc.

Standard	Application Areas
MPEG-1	VideoCD, MP3, videos on home computer
MPEG-2	DVD, digital television
MPEG-4	WWW, interactive television, mobile computing
MPEG-7	Content-based retrieval, content-description
MPEG-21	Digital Rights Management

Table 1: MPEG-standards and their applications

A user provided with an appropriate viewing device, could make the most of these novel techniques and relatively easily access background-information. Questions such as *"I wonder what the name of this actor is!"*, *"In which other movie was this actress starring?"*, or *"What is the title of the song in the background?"* could immediately and easily be answered. In the case of background music, for example, the user could even copy the sound-file to the hard disk if it is a separate object (MPEG-4) and the recording is not copy-protected and copyrights are not violated (MPEG-21).

4.3 MPEG-7 Details and Core Components

The official designation of MPEG-7 is "Multimedia Content Description Interface". It defines standardized elements, a standardized *interface,* for the description of content. Hence the scope of the standard does not cover the generation of a description. Neither automatic nor semi-automatic algorithms for feature extraction are explained or recommended; algorithms for search-engines that process MPEG-7 data are not mentioned either. These parts are clearly out of the scope of the standard.

4.3.1 Scope and Storage

MPEG-7 does not define a monolithic system for the description of the content like pre-defined fields in a database that have to be filled out. Such an approach would be limited to pre-defined schemes, might not include all existing applications and not take all possible future applications into consideration (e.g., [Wollborn 2001]).

The language of choice for descriptions is XML. They can be stored in the files' headers (e.g., PNG- or MP3-files) or are multiplexed on top of audio-visual streams. However, the descriptions do not necessarily have to reside attached to the data itself, they can also be retained in separate XML-files or databases, even on remote computer systems such as a central "MPEG-7 Description server".

4.3.2 Specification of Descriptions

The MPEG-7 standard allows for multilevel descriptions. In this way, features can be described on different levels of abstraction. On the lowest level are basic features such as the color of an object or the pitch of a tone. The structural level includes the representation of regions and segments. As depicted in figure 1, higher levels include semantic information or bibliographic details. The higher the level of representation, the poorer is the performance of automatic feature extraction algorithms, and the higher the level of human interaction involved.

The MPEG-7 paradigm relies on three elements of description: Descriptors (Ds), Description Schemes (DSs), and Data Description Languages (DDLs). A Descriptor represents a single feature, and therefore it can be seen as atomic element of MPEG-7 descriptions.

A Description Scheme defines the structure and semantics between Descriptors and other Description Schemes. Thus, a Description Scheme can contain Ds and other DSs. If a DS is represented as a tree, Ds cannot be inner nodes but only leaves (see figure 2).

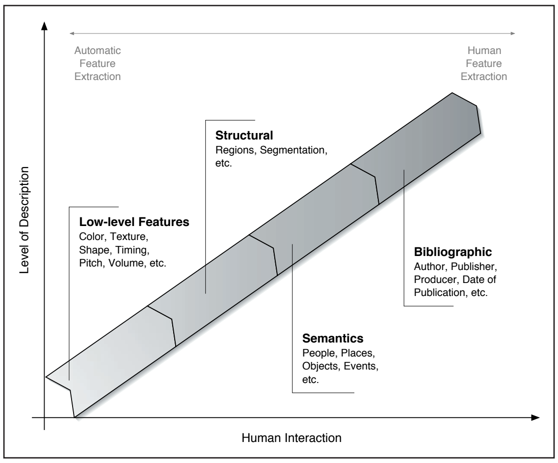


Figure 1: Multilevel descriptions and the application of automatic feature extraction.

The Data Description Language is a language for creating new Description Schemes and defining new Descriptors (i.e., features). It is also capable of extending and modifying existing DSs. The DDL validates the content and structure of Ds and DSs (see [MPEG-7 2002]).

The language of choice is XML with some MPEG-7 specific extensions including arrays and matrices. This has several advantages: XML is platform- and machine-independent and it is both human- and machine-readable. Furthermore, XML is already an accepted standard, and it can be validated easily. In case the textual coding using XML is too inefficient, a binary representation named BiM can be used. Among other benefits, BiM offers bandwidth-efficient transmission and great flexibility for streaming media.

4.3.3 Examples of Descriptors

This section briefly lists some of the descriptors that are currently available. Detailled illustrations can be found in [Hunter 2001] and [MPEG-7 2002].

For visual objects among others the following features and descriptors are defined:

- Color: color space, color structure, dominant color, scalable color;
- Shape: edge histogram, contour shape, region shape;
- Motion: object trajectory, camera motion, parametric motion;
- Texture: homogenous texture, texture browsing;
- Semantic: Face recognition.

Low-level audio descriptors include basic spectral or waveform and power values, parameters of the timbre, spectral basis, and silence.

Obviously, multimedia elements are a compilation of several simple media objects such as sound-objects or images. Therefore there is no multimedia descriptor, but a multimedia description scheme. The description scheme labelled DS1 in figure 2, for instance, could represent a multimedia object.

4.4 Prototypes of MPEG-7 Applications

Although MPEG-7 is a relatively new standard, some application prototypes are already available. IMKA ([IMKA 2001]), developed at Columbia University, and TV-Trawler ([TV-Trawler 2001]), proposed by the Distributed Systems Technology Center (DSTC) at the University of Queensland, are discussed in this section.

4.4.1 IMKA

Since both the creation and consumption of metadata are outside the MPEG-7 standard, a workgroup at Columbia University initiated projects that focus on these applications. IMKA, an "Intelligent Multimedia Knowledge Application", is one of these applications (e.g., [Benitez et al. 2001a]). It is a retrieval system for multimedia data.

IMKA relies on MediaNet (see [Benitez and Smith 2001] and [Benitez et al. 2000]), a knowledge representation framework that uses multimedia information in order to generate semantic and perceptual networks. This

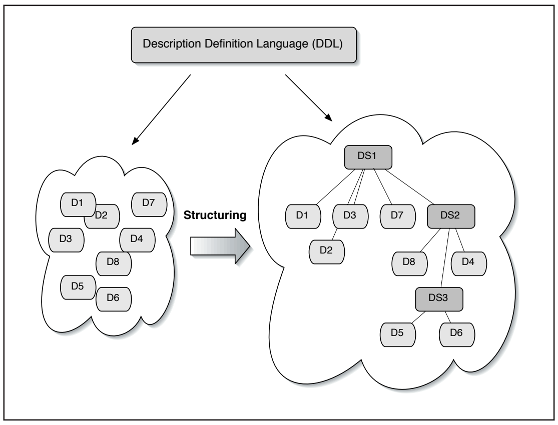


Figure 2: Descriptors (Ds) and Description Schemes (DSs) in a structured, hierarchical tree representation. The Description Definition Language (DDL) is used to validate Ds and DSs.

information comes from MPEG-7 descriptions and other annotations to pictures of an image collection. Moreover, low-level features such as the color histogram and texture information are created for the objects.

WordNet ([WordNet 2002]), an application similar to a thesaurus, is employed to establish relations such as is-similar-to or is-generalization-of among the nodes of the network in a semi-automatic process. Hence, a visual query for semantically or perceptually similar objects is made much easier.

In contrast to the next example, in IMKA MPEG-7 is not the primary concept that is employed (this is MediaNet), but only a model that enables and standardizes content- and feature descriptions.

4.4.2 TV-Trawler

The TV-Trawler project describes an application that is aimed at digital video broadcasts ([TV-Trawler 2001]). The system analyzes incoming video content, mainly digital television broadcasts, and matches it against stored user-preferences. If the two correspond, the video stream is recorded, or the user is notified of the event, or both.

The system can be split up into two components, a preparation phase and the actual action that is carried out. In a first step, a new user has to complete a user-profile that collects various information including preferred television broadcasts, favorite actors and directors, and certain programs and channels. This information is retained in a MPEG-7 compatible "User Preferences Description Scheme".

For all video content, the MPEG-7 descriptors have to be extracted from existing metadata sources, which are service information packets, closed captions, and electronic program guides (EPGs, see [Rogers et al. 2001]). Service information packets are an extension of the MPEG-2 standard that can contain details about the program in the digital stream. Closed captions, teletext, are a rich resource for extracting all kinds of metadata. The drawback is, though, that no standardized format or structure is available, which makes automatic extraction more difficult. The third origin of content descriptors are electronic program guides. These are basically television guides that are freely available on the world wide web.

The automatically extracted metadata are stored as MPEG-7 conformant descriptions.

The second component of the system is the filter engine. When the system encounters a potential interesting video broadcast, it notifies the user and, according to the user's preferences, records the program or automatically starts a device capable of handling the streamed content.

Since the system is targeted at digital television, most incoming data will be encoded using MPEG-2 DVB (digital video broadcast). However, the content is stored as MPEG-4 because this representation is more efficient and offers more features (see above). The recorded video documents are added to the user's video archive and tagged with the date, time, and additional data of the recording.

Although this feature is not explicitly mentioned, the user's feedback could be employed, to enhance the performance of the filter engine. If the user actually watches the recorded video broadcasts, the system could put emphasis on this kind of program, while other programs, that are clearly not appreciated by the user, could be removed from the user's preferences.

4.5 Other Metadata Initiatives

Many researchers and developers already see MPEG-7 as the standard for describing metadata and multimedia content. However, there are several other initiatives that are used at present and have yielded established applications. Two of them, the Dublin Core Metadata Initiative and the Open Archives Initiative, are presented in this section.

4.5.1 Dublin Core

The Dublin Core Metadata Initiative (DCMI, [DCMI 2002]) was founded in 1995 at a workshop in Dublin, Ohio. Its goal is to standardize and promote the development of specialized metadata vocabularies for describing resources. One intended purpose is the semantic description of WWW resources. This is done in a similar way as a library card catalog stores metadata about books in a library.

The Dublin Code Metadata Element Set (DCMES, [DCMES 1999]) was published as IETF RFC 2413 (see [RFC-2413 1998]). It is a very simple description scheme consisting of only fifteen depictive, semantic definitions. The following list enumerates the descriptors.

- Content: title, subject, description, type, source, relation, coverage;
- Intellectual property: creator, publisher, contributor, rights;
- Instantiation: date, format, identifier, language.

Due to its descriptive nature it is not restricted to a particular discipline but includes a broad variety of includes domains.

Dublin Core is available as a unqualified (simple) and qualified variant. The unqualified version, mentioned above, consists of <attribute, value> pairs. The qualified

Dublin Core, though, uses additional qualifiers to further refine a resource. In that scheme, a field might not be a simple <attribute, value> pair but a more sophisticated data structure.

The Dublin Core metadata descriptors can be stored in many different ways such as <META> tags in the header of an HTML-file, in separate RDF- or XML-files. Search engines such as Microsoft's Index Server and Verity Search 97 Index Server are ready to index the <META> tags defined by Dublin Core (e.g., [Powell 2002]).

Advantages of Dublin Core are simplicity of creation and maintenance, commonly understood semantics, availability in various languages, extensibility, and conformity to existing and emerging standards ([Hillmann 2001]).

Since the original Dublin Core project concentrates on textual content, an extension for describing digital video and audio content has been presented by the Video Development Initiative (ViDe, see [Agnew and Kniesner 2001]). The fifteen descriptors of the DCMES are used to define metadata of images, audio- or video-content. However, this approach only deals with the documents as such and not with the actual content. Further projects make use of the qualified Dublin Core in order to describe the content itself and even low-level features ([Hunter and Newmarch 1999]).

4.5.2 Open Archives Initiative

The Open Archives Initiative (OAI) identified the issue that in many cases metadata is readily available but that it is hard to get it out of the archives in which it is stored. One problem is that in different archives metadata is retained in different formats, and that not all systems deal with the same set of descriptors. Hence, the initial intent of the OAI was to unite the many different collections of mostly scientific documents, such as papers, proceedings, or technical reports. As the requirements grew, the OAI became a generic metadata transport protocol. It is independent of the type of document or media, and the kind of metadata.

The OAI Metadata Harvesting Protocol ([OAI 2002]) uses unqualified Dublin Core as the lowest format of metadata representation. Other, more sophisticated formats are possible, but have to be stored in structured, verifiable XML-files. The relation to Dublin Core brings all its advantages and also the disadvantages such as the primary focus on text documents.

Since the OAI project relies on Dublin Core, it can be understood as an application of Dublin Core. Although OAI *is* on a "higher level" than Dublin Core, it is not a practical application but a framework for standardizing and *implementing* software that has to deal with metadata and the description of content.

Hundreds of systems have adopted the OAI Metadata Harvesting Protocol, including many government organizations such as Nasa or universities such as the MIT (e.g., [OAI 2003]).

4.6 Conclusion

MPEG-7 is a very promising new standard that complements the MPEG-family. It specifies ways of defining metadata for all kinds of media without narrowing the choices of generation and consumption of these data. The MPEG-suite with MPEG-1, -2, and -4 for the representation of digital content, MPEG-7 for the content description, and MPEG-21 for digital rights management will present a complete package that allows for the production, transmission, handling, and management of digital media.

When one looks at the dominance of the MPEG compression-standards and the effort that was undertaken in defining the multimedia content description interface, MPEG-7 seems to be the first choice for describing metadata. However, other standards have already existed for several years and have, in contrast to MPEG-7, already produced commercial applications.

It should be noted that MPEG-7 is a rather complex, heavy-weight standard that is not very easy to implement. Dublin Core, on the other hand, is a light-weight framework that has already been implemented and used successfully by many companies and organizations. It is easy to understand, and uncomplicated to implement.

So although the MPEG-7 framework is very powerful it might be too complex, particularly compared with Dublin Core or OAI. Many experts see MPEG-7 as the new standard for content description, but the acceptance of implementors and users has to be awaited.

5 Digital Rights Management

5.1 Introduction

This chapter introduces Digital Rights Management (DRM), the methodology that is used to manage, control, and secure data and the access to services in any kind of digital information system.

Although several competing developments in digital rights management are under way at the moment, only selected standards will be presented. MPEG-21 as the probably best known advance in digital rights management. As one of the underlying technologies in Part 3 of this thesis, it is now discussed in detail. Moreover, XrML is characterized as a rights expression language in this chapter.

The standards that are discussed in this chapter can all be used to *define* rights. However, the actual *enforcement* of these rights is not considered to be important for later chapters of this thesis, and therefore it is not dealt with in this section.

5.2 Digital Rights Management

The term "digital rights management" in a general definition characterizes the digital management of rights. Neither the objects not the rights themselves have to be digital data. Usually, though, DRM deals with the digital management of digital rights that are attached to digital objects.

Since the field of DRM is very broad, it is frequently subdivided into several domains. These include the identification and description of digital items, the specification of digital rights and their management as well as technical aspects such as cryptography and copy protection.

Traditionally, an attempt has been made to make the conventional content distribution process over the internet secure (see [Fowler 2002]). Therefore most DRM initiatives have concentrated mainly on technical issues and have tried to provide security for documents and services by means of encryption. Technologies such as watermarking and asymmetric ciphering have been used to make sure that content can only be accessed by those who actually paid for it (e.g., [Cheung and Curreem 2002; Knight 2003; Piva et al. 2002; Välimäki and Pitkänen 2001]).

In more recent approaches, DRM comprises the complete life cycle of information from creation to transmission and consumption. Furthermore not only the document itself, but also issues such as the description, identification and allocation, trading, and tracking are considered.

5.2.1 Business Models

In this context, the management, distribution, and marketing of content with both its internal and external relations can be called a business model. It describes on which basis actors (see below) may exchange or use data. If a DRM system does not support the business model of a particular company, it is not suitable for this organization.

Classic business models include, for instance, rent, gifting, library loan, site license, pay per view, subscription, preview, and unlimited usage. The actors in these models are consumers, retailers, distributors, publishers, editors, agents, and authors among others.

A specific example is the music and record industry. The business model is based on the assumption that it has been too expensive or difficult for the majority of users to copy the recordings, i.e. CDs. Traditionally, large-scale theft was dealt with by introducing legal barriers, and small-scale theft was basically prohibited by technological barriers. However, the introduction of cheap and easy-to-use CD-writers and the development of MP3 and similar technologies undermine this premise and render it inappropriate (e.g., [Cherry 2001; Cherry 2002; Zacks 2002]).

In this case, the mere application of technologies such as copy protection is not sufficient because the actual problem stems from the infrastructure that was used up to now to protect intellectual property rights. The business model as such is no longer suitable and will have to be adapted to the use of DRM technology and DRM systems. (See also [Erickson 2002; Iannella 2001] and [Davis 2001].)

5.3 MPEG-21

In the family of MPEG standards (see section 4.2), digital rights management is referred to as *Intellectual Property Management and Protection* (IPMP). MPEG is working on a standard for IPMP at the moment; its designation is MPEG-21, ISO/IEC 21000 ([MPEG-21 2002]).

5.3.1 Historical Background

The MPEG-21 specification has its roots in the IPMP initiatives of MPEG-4 (see Koenen 2001b,c]).

At a very early stage of the development of MPEG-4 DRM issues were encountered, and hence DRM/IPMP-features are built in MPEG-4 (e.g., [Koenen 2001a]). MPEG-4 includes both a descriptive section for defining attributes such as the author, copyright, and the type of content, and a section about content protection for building secure MPEG-4 delivery systems.

The prime deficiency is that MPEG-4 does not focus on IPMP, and the implementation of rights management is not mandatory. Only so-called "hooks" are provided, viz., standardized interfaces where proprietary IPMP components can be tied in. Thus, the piece of software that actually handles the rights management is not covered by MPEG-4 and does, in fact, not have to follow any standard.

However, interoperability and compatibility are one of the most important aims of the MPEG-standards. Since MPEG-4's IPMP is too loosely defined, many different (incompatible) but still similar devices could be introduced by different manufacturers. A consumer might, for example, buy a CD with protected MPEG-4 content and cannot use it because the player device does not support the required IPMP implementation. This makes "internetworking" impossible (e.g., [Erickson 2001; Mooney 2001]).

That concern was mainly emphasized by the SDMI, the Secure Digital Music Initiative. The SDMI anticipates an efficient copy-protection mechanism that does not restrict interoperability and internetworking: content should be protected but be as simple to use as conventional CDs.

As far as the implementation of MPEG-4 is concerned, the IPMP data is multiplexed with the actual content and is transmitted within the data stream. The MPEG-4 player demultiplexes the stream, finds the corresponding IPMP system, and processes the data. The IPMP component could be built in or could be available in the form of a plug-in to the player application.

The development of the MPEG-4 IPMP framework provided valuable experiences and the insight that a strong demand for a separate standard for digital rights management existed. That finally led to a new standard – MPEG-21.

5.3.2 Overview of the MPEG-21 Standard

MPEG-21 is a standard that contains nine parts, of which the most important ones will be discussed in the next few sub-sections.

First, the core concepts are briefly introduced. Then, the Digital Item Declaration (MPEG-21 Part 2) as one of the central technologies of MPEG-21 is described. Further sections explain the Digital Item Identification (MPEG-21 Part 3), and the Rights Expression Language (MPEG-21 Part 5) together with the Rights Data Dictionary (MPEG-21 Part 6).

5.3.3 Core Concepts

The MPEG-21 framework relies on two fundamental concepts: users and digital items. *Users* are individuals that act upon content (create, process, transmit, consume, etc.). They are usually people such as authors, editors, publishers, distributors, and consumers.

Digital Items, on the other hand, are objects such as text documents, images, drawings, sound files, and videos, i.e. they are content. Digital rights are always related to digital items.

Section 5.3.4 describes the Digital Item Declaration, the model used to define digital items, in greater detail.

5.3.4 Digital Item Declaration

Digital items are an essential part of the MPEG-21 standard, because they are the objets the DRM system deals with. In MPEG-21 a digital item is understood as "the digital representation of 'a work'" (from [MPEG-21 2002]). Therefore it is not necessarily a single document. An example is a book published on the internet as a collection of several HTML documents and images. The book as a work is a digital item. That makes it necessary to determine all resources that are associated with a digital item in an unambiguous way.

The Digital Item Declaration (DID) is the model in MPEG-21 that provides a set of elements and concepts for specifying digital items in an appropriate and exact way. It does not specify a directly applicable language but rather describes an abstraction on which a language (e.g., the rights expression language) can be based. The grammar of the DID is represented in an XML schema definition to facilitate reuse and extensibility.

From an object oriented point of view the DID is a construct of sub- and super-classes. This means that the declaration of items can be made on different levels of granularity. The top-level entity and most general construct is a *container.* It is made up of other containers or *items* and is used to build logical *packages* for transmission or exchange.

An item consists of other (sub-)items as well as *components* and *descriptors* (see below). As such, an item is only a structure for organizing and holding together the elements it contains. An example for an item is a presentation that contains background music.

Components define relationships between *resources* and their descriptors. Commonly, a resource is an unambiguously identifiable object such as a simple text document or an image. Its characteristics, for instance the image resolution or a preview image, are defined by descriptors. Descriptors can also be used to specify the features of other sorts of elements including containers and items.

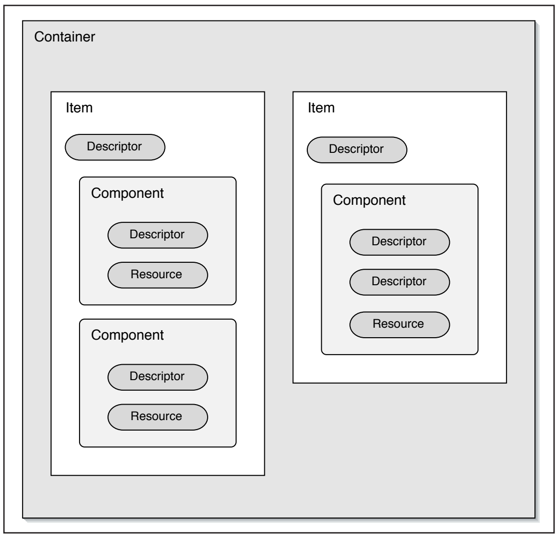


Figure 1: A simple example depicting the use of the most fundamental elements of the MPEG-21 DID: containers, items, components, resources, and descriptors.

Further entities specified by the DID include anchors, annotations, assertions, choices, conditions, fragments, predicates, selections, and statements.

The container in figure 1 encloses two items. The first item contains a descriptor and two elementary components, as opposed to the second item that contains a descriptor and one somewhat more complex component consisting of three descriptors.

5.3.5 Digital Item Identification

The Digital Item Identification (DII) specifies methods to uniquely identify digital items and their contents, intellectual property information, description schemes, etc.

MPEG-21 as a multimedia framework does not define any new identification or description schemes but supports and encourages the use of already existing technologies. A well known way of uniquely identifying objects is, for instance, the use of uniform resource locators (URLs) utilized on the internet. URLs are only a subset of the so-called "uniform resource identifiers" (URIs). All major, established identification systems including ISBN for books and ISRC for recordings can be understood and utilized as URIs. Some supported identification systems are listed in table 1.

URIs can either be references to external resources (on the internet, for instance) or to internal resources such as another item or component.

Identification System	Identifier
Content ID Forum (cIDf)	cid
Digital Object Identifier (DOI)	doi
EAN/UCC	ean *or* ucc
International Standard Audiovisual Number (ISAN)	isan
International Standard Book Numbering (ISBN)	isbn
International Standard Serial Number (ISSN)	issn
International Standard Recording Code (ISRC)	isrc
International Standard Text Code (ISTC)	istc
International Standard Musical Work Code (ISWC)	iswc
Version Identifier for ISAN (V-ISAN)	visan
Music Industry Integrated Identifier Project	mi3p

Table 1: A few identification systems that can be used with MPEG-21 (see [Sakamoto 2002]). Note that all systems except for cIDf, DOI, and MI3P are defined by the ISO.

The URI for uniquely identifying the book *"A Clockwork Orange"* by Anthony Burgess in MPEG-21, for example, would read urn:mpeg:mpeg21:diid:isbn:0-14-118260-1. (The term "diid" stands for "Digital Item Identification and Description".) This approach uses XML namespaces as mechanism to distinguish between identifiers of different systems.

As far as the storage of the DII data is concerned, the structures defined in the DID are employed. All identifiers that are specified as part of the DII are expressed with <statement> tags within the descriptors of the DID. Since descriptors can be attached to DID elements such as items and resources (see above), the DII information can also be appended to them.

5.3.6 Rights Expression Language and Rights Data Dictionary

A Rights Expression Language (REL) is a machine readable and at least semi machine enforceable language for declaring rights, permissions, and conditions. It is the language that actually describes the permissions associated with a digital item.

MPEG-21's REL is based on XrML 2.0. Its basic concepts are similar concepts, and it adopts the fundamental structural elements: principals, rights, resources, and conditions. For a more detailed description of XrML see section 5.4 below.

The Rights Data Dictionary (RDD) provides a set of precisely defined *terms* that are used by the REL to express rights. Hence, there is a close realtionship between the REL and the RDD.

The RDD of MPEG-21 does not only recognize the terms defined in its own dictionary but can also handle definitions from other authorities. This is done by mapping them from the external namespace into the local dictionary. In addition to this, the RDD can also transform and map metadata from a different namespace into the local one.

Obviously, the interoperability between different systems can be enhanced by a well designed RDD. Therefore one of the aims in designing a RDD is that it should be simple and unambiguous. Furthermore the transformations from one namespace to another one should be done fully or partially automated with a minimum of loss in detail and expressiveness.

5.4 XrML

XrML, the eXtensible Rights Markup Language, is a language for expressing digital rights ([XrML 2001; Miron et al. 2001]). It was originally developed at Xerox PARC as the Digital Property Rights Language, "DPRL" (1996-1999), with a focus on machine enforceable digital rights. After XML was introduced in favour of a Lisp-style language for representing the rights definitions, DPRL was renamed XrML, and the development was handed over to a newly formed company, ContentGuard.

Recently, XrML was accepted by the Oasis Rights Language Technical Committee as basis for creating a new, open digital rights language. Therefore, ContentGuard surrendered the development of XrML to the Oasis Group.

So far, XrML has not been an open, freely available standard because several early patents are held by ContentGuard. This situation might change in the future, though.

Among the users of XrML are HP, Microsoft for its eBooks, Sony, Time Warner, and Xerox.

Strategic alliances and cooperations with other organizations such as MPEG (see above), the Oasis Open group, TV-Anytime or the eBook Rights and Rules Working Group (RRWG) make XrML the currently most important and the most widely accepted rights expression language.

5.4.1 Design Goals of XrML

XrML was designed to be a language that can be used for all content- and media-types, on all platforms (software and hardware), with all kinds of formats, resources, and products. To put it a different way, *interoperability* was a chief concern. Moreover XrML should be independent of the domain and the application but still be able to describe a wide variety of business models. Furthermore, the language should be comprehensive, precise, and unambiguous.

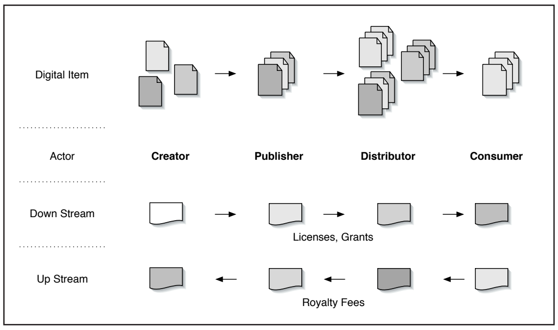


Figure 2: The multi-tier model used by XrML with up stream and down stream data flows. Redrawn from [XrML 2001].

It should be noted that the use of XrML is not strictly limited to DRM systems, but it can also be employed in content management systems, digital libraries and archives, systems with trust relationships, etc.

The XrML standard attempts to cover all media and production processes that are currently of interest. In order to minimize the cost of future changes, extensibility and adaptability are core features of XrML.

5.4.2 Business Models in XrML

As already mentioned above, the support of a wide variety of business models is very important for a DRM system. In fact, the crucial factor is the rights expression language. If the REL is unable to describe the conditions and permissions used in a business model, it cannot be used in this context. In general, this means that any DRM system based on this particular REL is inappropriate for that kind of business model.

XrML has addressed this issue and attempts to be as generic as possible. It provides a so-called "multi-tier model" illustrated in figure 2. This approach supports the independent specification of rights both up stream, from the consumer to the creator, and down stream, from a creator to the consumer.

Figure 2 shows the four most common actors in the life cycle of a digital item: creator, publisher, distributor, and consumer. The rights that are defined down stream usually become more and more specific and increasingly restrictive. In this direction licenses and grants are transferred while royalty fees are passed on up stream.

5.4.3 Structure of the Language

XrML is based on XML and on the XML Schema technology, hence offering great flexibility, extensibility, coupled with robust expressiveness.

The language in its current version 2.0 (see [XrML 2002]) consists of the core schema with the most elementary concepts of the language, and two built-in extensions, the Standard Extension Schema and the Content Extension Schema. Domain specific extensions can be freely defined and added to XrML. The newly defined extension schemas can, in turn, be extended by further extensions.

5.4.4 Basic Components of XrML

The XrML data model consists of four basic entities: principals, rights, resources, and conditions. A *right* describes an action that is granted to *principals* such as content owners or distributors, in order for them to be able to access a *resource*. In general, a right describes an action that a principal may perform on a resource, e.g. view it, modify it, or transmit it. A resource is an object to which rights are attached. It can, for instance, be a document, any other piece of information, or a service. The *condition* determines under which circumstances the rights are actually granted.

These four entities are used to form more sophisticated structures such as grants and licenses. A *grant* is the element that authorizes a principal to use or pass on a resource. A set of grants together with one or more *issuers,* and optional information compose a *license.* An issuer is a principal who issues a license, hence giving grants to another principal. The structure of a license is depicted in figure 3.

A grant and its four core components can roughly be seen as the basic elements of an English sentence: subject, predicate, object, and adverb. The principal is the subject, the predicate is a right that is granted, the object is a resource, and the condition resembles an adverb. An example: the user (subject) has the right to read (predicate) the book *"Alice's Adventures in Wonderland"* by Lewis Carroll (object) once (adverb).

5.5 Related and Other DRM Initiatives

MPEG-21 and XrML are not the only DRM initiatives under way. Virtually dozens of competing standards have emerged within the last two years. Some of these projects are briefly outlined together with related topics in the following sub-sections (e.g., [Rein 2002]).

5.5.1 IDRM

The Internet Digital Rights Management (IDRM) Initiative by the IETF (see [IDRM 2002]) is not a competitor of MPEG-21 or any other DRM system. Its task is rather to analyze the impact of DRM on the internet as well as on its architecture and structure. Furthermore it seeks to investigated how open networks such as the internet might have to be adapted in order to support DRM ([Hardjono and Baugher 2001]). The group also wants to encourage the IETF and the IRTF (Internet Research Task Force) to adjust their engineering and developments to the needs of DRM.

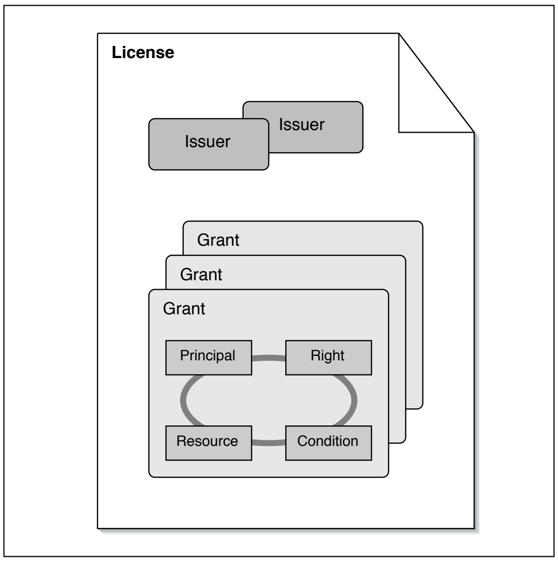


Figure 3: An XrML license consisting of one or more grant(s) and one or more issuer(s). The fundamental elements of a grant are principals, rights, conditions, and resources. Based on [XrML 2002].

Therefore several cooperations within the IETF (with other workgroups) and external organizations such as the W3C or the Digital Object Identifier (DOI) initiative exist. A liaison with the MPEG-21 workgroup has been formed, and other standards such as ODRL or XrML are surveyed as well (see [IDRM 2001]).

5.5.2 ODRL

ODRL, the Open Digital Rights Language, is a combined REL and RDD developed by IPR Systems in cooperation with Nokia and RealNetworks. As such it is a direct opponent of XrML. It is used and supported by a number of companies including Adobe, IBM, and Panasonic, and organizations such as Vienna University of Economics and Business Administration ([Iannella 2002; ODRL 2002a,b]).

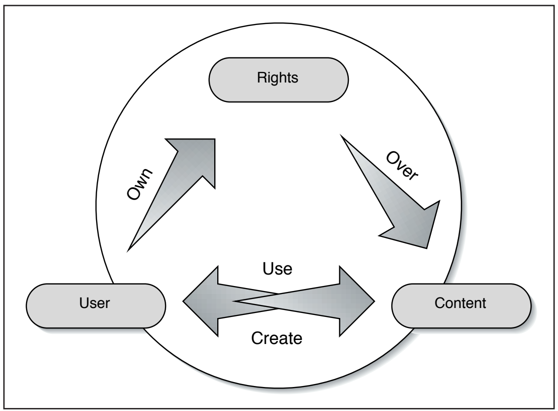


Figure 4: The ODRL model. Users own rights over content. They create new and use existing content.

Unlike XrML, ODRL is a freely available open source standard. The specifications of the language can be downloaded free of charge, and no license is required.

Roughly outlined, ODRL defines users, rights, and content as basic elements of the language. Users are individuals such as authors, distributors, or consumers. They can create new content or use – edit, process, distribute, transmit, etc. – existing content.

The rights that are owned by users are made up of items such as signatures or permissions containing constraints, requirements, and conditions. Permissions express the actual right to access, view, print, etc. the content. The optional constraints are restrictions to these rights. An example would be that the user is allowed to view the content (permission) but only in a lower quality (constraint). Requirements are prerequisites that have to be met before the permissions are granted. Users could, for instance, be required to pay (requirement) before they are allowed to print the content (permission). Finally, conditions are exceptions that have to be false for the permission to be valid. If a condition becomes true (e.g., a credit card expired), the permission will become invalid.

The ODRL information model depicted in figure 4 consists of only three entities as opposed to the XrML model that uses four. The missing element is the condition that is a part of the rights entity in ODRL.

Like XrML, the specification of ODRL makes use of the XML schema technology: one schema defines the REL, another one is needed to specify the RDD.

In general, XrML and ODRL tend to be quite similar which is only natural because both the purpose and the goals of the two languages are congruous. The only big difference, as already mentioned above, is that ODRL is an open source development, while XrML follows a rather strict licensing policy.

5.5.3 cIDf

The Content ID Forum, cIDf, is a Japanese initiative founded 1999 at the University of Tokyo. The intent was to create "a framework for contents distribution promotion while protecting copyrights" (from [Sakamoto 2001]). As of 2001 more than 130 (mainly Japanese) organizations and companies were members of the forum.

The cIDf specification covers not only the identification of digital items but also parts of the DID and RDD sections in MPEG-21 as well as some basic metadata facets (see [cIDf 2002]). The content ID that is defined for each object consists of two parts: the work ID and the issuing ID. The work ID is created when an object is created. It contains a unique identifier for the object as such and attaches a set of metadata including content characteristics and rights attributes. In contrast to this, the issuing ID is generated at the content distribution.

This is a feature that distinguishes cIDf from most other approaches: both static (work ID) and dynamic (issuing ID) attributes are used to identify content. This situation is intended because it is considered to be important to identify content on basis of its distribution conditions.

An example: a user has the option of viewing a video over a high-bandwidth network connection (higher quality) or a slower modem connection (low quality). In this case, the distribution channels are different and therefore the issuing ID will be different. This approach is quite sensible because different copyrights and different fees might apply.

The metadata that is collected and stored in the database is publicly accessible. Users are is encouraged to query it in order to obtain information such the author of a document or the copyright record. This is rather different from the

ISO systems such as ISAN or ISBN where the database is only available for internal use.

Security plays a central role in the cIDf system. A two-layer digital watermark, for example, is included to protect the content ID. This makes it possible to also check its authenticity and integrity. In addition to this, optional fields for digital signatures and further measures are available. Interestingly, cIDf describes in this context a so-called “net watcher” application, a software program that searches networks for illegally copied content and tampered content IDs.

A distributed database systems is employed for the storage of the cIDf data. A practically unlimited number of Content ID Management Centers (cID-MC) can be set up. Each center is identified by a unique number. This ID is the first part of the object identifier. The second part of the object ID can be freely assigned by the cID-MC. The only requirement is that it has to be unique within the given cID-MC. This technique makes it easy to incorporate other content identification standards in cIDf. If the authority managing the ISBN were a cID-MC, for instance, it could simply reuse the existing ISBNs: the first part of the object identifier is the (unique) cID-MC number, the second part is the existing ISBN.

cIDf seems to be a modern, well designed standard with an academic background and strong commercial support. Aspects such as security and reuse of existing numbering schemes make it a sound technology.

5.5.4 DOI

The Digital Object Identifier (DOI) Foundation, established in 1998, is an international organization based in Geneva and Washington. Its focus is on providing a system for the “persistent and actionable identification and interoperable exchange of intellectual property in digital networks” (from [Paskin 2002a,b]).

Currently over 200 organizations, mainly in the (print) publishing sector, use the DOI system to identify the objects they produce and process.

A digital object identifier consists of two parts: a prefix and a suffix; they are separated by a slash symbol. The prefix designates the organization that issues the DOI. The suffix is a code that can be freely chosen as long as it is unique to the prefix. The approach is quite similar to the cIDf system and makes it possible to embed IDs assigned by other authorities into the DOI system. This enables interoperability.

The DOI Syntax was approved as an ANSI standard (ANSI/NISO Z39.84-2000) in 2000.

The DOI initiative uses a resolution technique to associate every DOI with an actual resource like a URL. Once a DOI is attached to an object it remains persistent, i.e. the DOI number will not be modified if information such as the ownership or the location of the object changes. The users operate with the same DOI, and the resolution system points them to a different location or returns a different ownership record ([Paskin 2002c]).

Apart from that, DOI numbers are “actionable”. They can be compared to a traditional phone number: the number does not only stand for an address, but it can be used to *do* something, to establish a connection with another telephone. Similarly, a DOI number can be used to locate the actual object on a computer network (with the help of the resolution system) or to present the associated metadata (see [DOI 2003]).

The metadata set employed by the DOI Foundation is based on <indecs>, a metadata framework for interoperability of data in e-commerce systems (see [Rust and Bide 2000]). It is compatible with the MPEG-21 RDD and allows for mappings between different application domains.

DOI is quite different from cIDf in that it defines static, persistent IDs, whereas cIDf makes use of a dynamic component to generate its IDs. However, one cannot determine which approach is “better”. This strongly depends on the particular application or business. eBooks, for example might probably rather be tagged with DOIs, whereas video streams could potentially make use of the dynamic nature of content IDs.

5.6 Conclusion

The introduction of DRM will have an impact on the way content is disseminated on computer networks. DRM systems will be launched to fulfill the content authors’ and publishers’ demands and to make intellectual property laws enforceable on the internet.

MPEG-21 is the MPEG’s attempt to address these issues and to propose a new multimedia framework for intellectual property management and protection. The MPEG-21 comprises DII, DID, and REL in different parts of the specification.

Certain domains such as the DII or the REL are also specified by many other organizations. From the great number of object identification standards two more recent developments, cIDf and DOI, were presented. They have different roots and varying priorities.

Also many different and yet similar RELs are available. In this chapter two of them, XrML and ODRL, were demonstrated. Both have support from notable organizations, and most likely both of them will play an important role in the future.

However, as far as MPEG-21 standard is concerned, a prediction of its acceptance cannot be made. Although a big effort is made to “conceive” this all embracing multimedia framework, it is still under development. Hence, its usability and effectiveness are still to be verified.

Part 2

Extensions to Active Documents

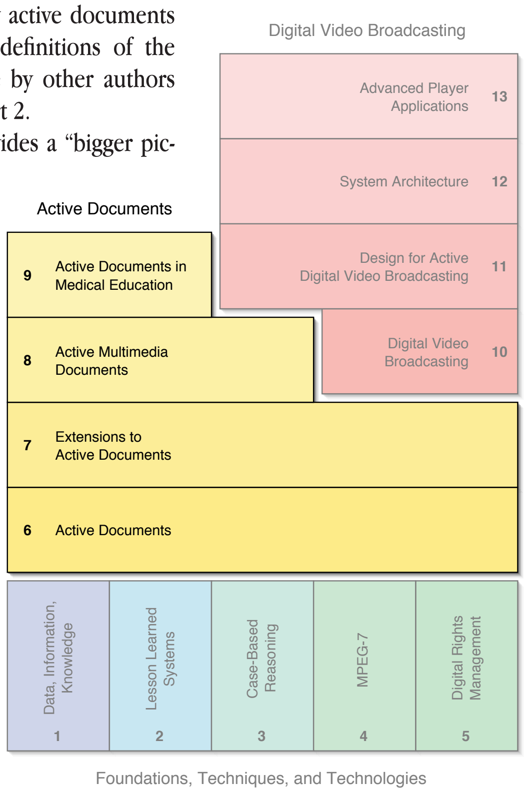
Extensions to Active Documents

The second part of this thesis addresses active documents. They serve as one of the fundamental concepts on which the KM system proposed in Part 3 is based.

Chapter 6 introduces Maurer's idea of active documents. First, the technique is described together with its limitations and some requirements that have to be met. Then, the architecture of a system utilizing active documents functionality is explained, and the details of an implementation are described. Additionally, several examples of systems that actually employ active documents are provided. Three disjunct definitions of the term "active documents" made by other authors conclude the first chapter of Part 2.

The subsequent chapter provides a "bigger picture" of the use of active documents. It presents several techniques that complement the functionality of active documents in digital libraries. Moreover, this chapter details how active documents can be seen as a valuable source of data in knowledge management systems.

Although the notion of active documents is not strictly limited to text-based systems, their use in media-rich environments has not yet been discussed. Chapter 8 extends the basic concept of active documents for the use in the multimedia domain. First, several supplementary and alternative input methods are described: region selection, texture and pattern descriptors, the query-by-example methodology, motion sketches, and speech and music descriptors can be used to express active documents queries. Based on these techniques, active multimedia documents are introduced. Active multimedia documents apply the active documents functionality to all kinds of multimedia objects: drawings and vector graphics, photos and rasterized images, video content, sound and music, and compound documents. Finally, some aspects related to the representation and storage of the information that is collected in connection with active multimedia documents are outlined.



A practical application of active multimedia documents is shown in chapter 9: ADIME, a concept for active documents in medical education, is presented.

ADIME is a concept that can enhance e-learning systems in media-rich environments. It focuses on the medical domain because this field heavily relies on images and video content. As an introduction, the current situation of e-learning systems in medical education is outlined. Then, the basic idea of ADIME is described, and the prototype of a user interface is presented. Furthermore, implementational aspects and details related to the active documents functionality are described.

This chapter concludes Part 2, *Extensions to Active Documents.* The third part of the thesis makes use of active multimedia documents and introduces a design for active digital video broadcasting.

Active Documents

6

6.1 Introduction

This chapter introduces the idea of active documents. The term "active document" has been used by many authors, and a variety of different definitions exists. Therefore the meaning used in this thesis has to be clarified.

The first section of the chapter presents active documents as they are specified by Maurer. The architecture of a generic system realizing the concept is explained together with some other details of an implementation. Subsequently, some examples are described.

The concept of active documents by Maurer is the definition that will be used throughout the remaining chapters of this thesis.

The last section of this chapter outlines various other projects that introduce different definitions of "active documents." Although these approaches have rather divergent directives, they pursue the common aim to make documents more active.

6.2 Active Documents

The idea of active documents was introduced in [Heinrich and Maurer 2000]. Active documents are an abstract concept that can basically be applied to any type of document. The idea is that users can ask arbitrary questions to documents, and answers are provided immediately and seemingly by the document itself.

The implementation of this idea includes an "online" and an "offline" component. When a question is asked by the user, the active document system (see section 6.2.2) checks if a question that is semantically equivalent has been asked before. If one can be found its answer is presented to the user. Thus, the answer is provided "online."

If an appropriate, existing question cannot be retrieved the user's request is forwarded to a human expert, and the user gets an apologetic message that an answer will be provided as soon as possible. In this case, the answer is provided "offline" by an expert. Alternatively, an answer can also be given by another user of the system.

In the course of time, answers are available for the most significant and most frequently asked questions. Therefore the human experts are no longer required, and answers can be provided by the online component. The phase in which experts have to be employed can be seen as a transient phenomenon.

6.2.1 Requirements and Limitations

The idea and implementation of active documents is based on two assumptions. The first one is that significant documents are accessed by a large number of users. The second observation is that the number of new questions decreases rapidly after a certain number of users has read a document.

This lets us formulate a "convergence criterion" that serves as a requirement for an active document system: in order to work effectively, the system should have a large user-base and a relatively limited set of documents. In this case, after some n users have submitted questions hardly any new questions occur, and the system can operate without experts.

If the convergence criterion is not met, i.e., there are too many documents or too few users, new questions are asked frequently. Hence, experts are kept busy, and the system cannot reach a stable state to work efficiently.

Systems realizing this idea of active document have successfully been implemented and have been used for some three years (see section 6.3 below). Their usage basically confirms the stated convergence criterion for a number n of 500 to 1,000 users per document.

One of the limitations of active documents is that they are only valid for rather static information. Documents that are created dynamically or whose validity is limited in time can ususally not make use of active document features. On a dynamically generated web-page that shows current money exchange rates, a user's question might not be sensible because it would only refer to this very instance of the page. The question and the corresponding answer might not apply for previous or future versions of the same page. Similarly, on a web-site about general computer hardware an answer to the question *"Which computer is the fastest one?"* is only valid for a certain time.

This problem can partly be solved by choosing answers very carefully. An answer referring to the fastest computer, for example, could not name a specific model but rather point the user to a document that provides up-to-date information on current computer hardware.

Another issue is that it might sometimes be better to change the actual document than to attach question-answer pairs; in some cases it is more efficient to redesign the structure of a document and to clarify explanations. However, active documents provide the means to detect which documents potentially have to be changed, and which parts of a document need to be improved.

6.2.2 System Architecture

Active documents can be seen as an add-on to traditional documents and services handling documents. Thus, virtually any document can be enhanced with active document features, and basically any kind of system – from a simple web server to an integrated document management system or an e-Learning enviroment – can be extended with active document functionality.

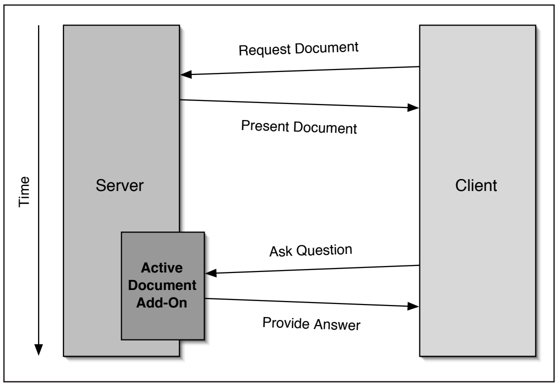


Figure 1: Architecture of a typical active documents system.

On the client-side, the active documents add-on provides the user with an interface to ask questions to a document. The server-side component receives the user's question, tries to find the corresponding answer, compiles a response, and sends it back to the client.

The communication model is outlined in figure 1. First, the user views a requested document. This functionality is provided by the unmodified "passive" document system. When the user asks a question, the resulting operations are handled by the active documents add-on: the component attempts to answer the question and finally sends a response back to the client. Without the active documents component and the arrows labeled "ask question" and "provide answer" figure 1 would show a usual, passive client-server communication architecture.

6.2.3 Details of an Implementation

The key component in an implementation of such an active document system is the module that checks if two questions are similar or semantically equivalent. [Heinrich and Maurer 2000] describe three methods to implement this fundamental module: a linguistic, a heuristic, and an iconic approach.

LINGUISTIC APPROACH. The system lets the user type in a question in natural language. Subsequently, the system checks if the entered question is semantically equivalent to a prior one. The problem is, however, that this process requires the system to understand natural language, and therefore it is, practically speaking, not yet applicable.

One approach to resolve the complexity of the issue is to restrict the syntax of questions to a basic set of terms that can be utilized by the user to formulate queries. A drawback of this method is, though, that the user would have to know the vocabulary that is available. Recent research in the area of learner-formulated natural language queries includes, for example, [Heinrich et al. 2001].

HEURISTIC APPROACH. The heuristic approach attempts to find similarities between new questions and existing ones based on heuristics. Methods from simple word matching to syntactic analysis and semantic networks (e.g., [Miller 1993]) can be employed for this purpose.

When a similar question can be found in the system, users are asked if the stored question matches their original one. If so, the retained answer is displayed. Otherwise the question is forwarded to an expert, and the user gets an apologetic response from the system.

The implementational complexity of this approach varies depending on the heuristic method that is chosen and can range from straightforward (word matching) to relatively complex (syntactic analysis). For users, this concept is advantageous because questions can be asked using natural language. Moreover, no special knowledge (such as the usage of a restricted grammar or syntax) is required to be able to utilize the system.

ICONIC APPROACH. The user asks a question by selecting a section of the information that is displayed on the screen. This means that the user's question refers to the marked part on the screen.

After the question has been answered, an icon is attached to the piece of information in order to make other people aware of the fact that an annotation is available. If other users have questions concerning the same section, they click on the icon first, in order to obtain a list of questions (and answers) that are already accessible.

One advantage of this approach is that it is relatively easy to implement because only the "location" of the icon together with the question-answer pair has to be stored in a database. Stochastic methods or AI-techniques are not required to find out if questions are similar or equivalent. Furthermore, it is easy for users to recognize, where questions have been answered so far, and which sections seem to have been unclear to other users.

6.3 Examples for Active Documents

Examples for active documents can mainly be found in document management systems and web-based training environments. The "Gentle" WBT-system as well as the Hyperwave eLearning Suite, for example, include active document features (see [Hyperwave 2003; eLS 2003] and [Dietinger and Maurer 1998; Gentle 2003]).

Especially in e-learning systems, active documents are a valuable tool. Learners can ask questions in the online environment while they are reading a document. They don't have to search for an e-mail address or contact the lecturer via a different medium. Users simply ask their questions and they are answered automatically or forwarded to the appropriate person, thus saving time and overhead.

Further applications of active documents include software support systems and digital libraries. Some examples are outlined in the following sub-sections.

6.3.1 Digital Libraries

According to [Maurer 2001], digital libraries should not be seen as static information repositories, but should much rather be seen as dynamic archives that let users interact with content and other users. Central aspects in such an environment are the ability to attach annotations even to small pieces of information and active document capabilities.

Annotations are well known from many digital systems: users can annotate PDF documents, attach (simple) comments to JPEG images, and Mac OS, for exmaple, lets users add annotations to any kind of document or application program. Although many digital libraries have not offered ways to let users add comments, some modern systems even allow for different levels of annotations: private annotations (can only be accessed by the owner), annotations that can be accessed by a certain group of users, and publicly available ones (e.g., [Maurer 2001]).

The functionality provided by active documents supplements and enhances traditional annotations. The active document approach is particularly well suited for digital libraries because:

- the retained documents are static;
- documents usually remain in the archive for a very long time;
- a large number of users accesses the documents, which facilitates meeting the convergence criterion.

A system that implements the functions described above is, for instance, the Journal of Universal Computer Science (see [JUCS 2003]). Further examples are provided in section 7.2.

6.3.2 Software Support

When software companies release new products, for example, consumers are often asking similar questions or are reporting problems that refer to a certain function that is described in a particular section of the online manual. In a conventional system, the software support staff would have to look up the manual, search for the corresponding paragraph, and probably find out that the problem has already been addressed before. Active document features can remove these redundancies: once the question is answered by an expert, other users get a system-generated response (linguistic and heuristic approach) to their request. Alternatively, users can see that an annotation to their problem already exists (iconic approach) and they don't even have to post a new query to the system.

In general, this approach can not only be applied for software manuals but for all kinds of user guides that are available in networked computing environments.

6.3.3 Software Products

Features of active documents could also be integrated in application programs. Imagine a help system, for example, that lets users ask questions that are related to help topics. A number of pre-determined question-answer tuples is installed together with the conventional help system. When the user asks a question similar to a stored one, the answer can be provided immediately. If a question occurs that is not covered by the local database, and the user's computer is connected to the Internet, the request is forwarded to an active document server. The server handles the request as explained above and can ideally provide a satisfactory answer within seconds.

This approach has another advantage: the software company gets valuable feedback from the user. This data can be used to enhance the product, viz., redesign the user interface or make some functions more intuitive. Moreover, the quality of the help system can be increased.

6.4 Other Definitions to Active Documents

The literature knows several definitions of active documents that are different from the one explained above. They originate from as diverse areas as property-based document management systems and ubiquitous computing, and therefore they pursue distinct aims.

Some of these concepts are briefly described in the next few sub-sections. Since all of these projects use "active documents" in their title, the sub-sections are named after the organization that carries them out.

6.4.1 Xerox PARC

The Xerox Palo Alto Research Center has observed that in current computer systems documents are stored in a particular place such as a folder in a hierarchical filesystem, and tied to these locations, there are often certain functions including access control and backup facilities. Moreover, the place where a document is stored very often also carries some semantic information. The location of the file /home/josef/papers/iknow-03/draft.pdf suggests that the document belongs to Josef, and that it is a draft for a paper to be submitted to the iKnow '03 conference.

Xerox makes an attempt to reform this paradigm and developed a document management system for more active documents called "Placeless Documents" (e.g., [Dourish et al. 2000]). The approach of the Placeless Documents system associates a set of attributes and functions *directly* with the document. So the semantic information that the document is a draft of a paper, that it is determined for a certain confernece, and that Josef is the owner is stored as metadata. However, also functions can be attached to the document. A common function is a backup functionality, for example.

In this approach, the document *itself* is responsible for being backed up. Another useful function is a notification mechanism. Whenever the document is modified, a group of users can be notified about the changes. Note that the document *itself* takes care of the notification.

This means that the document as such is taking a more active role. Its position moves from the passive piece of information that is processed by application programs to an active component in a computer environment.

6.4.2 Stockholm University

A somewhat different approach is made by the Research Group on Ubiquitous Computing at Stockholm Univeristy. The project demonstrates a concept of active documents that is designed for the use in ubiquitous computing environments (see [Werle 2000; Werle and Jansson 2000]).

The model is based on the following assumption: users require different documents when they are in different locations or meet different people. For certain tasks, it is also necessary to have only a certain *view* of a document. Therefore a system with the following characteristics was designed: documents try to find out in which context a user currently works and whether they are required. If a document might be useful in a certain situation, it (the document!) tries to find out which parts of the content are relevant, and forwards them to the particular user(s).

The proposed system is quite complex and includes a number of processes. One service, for instance, checks periodically which people are logically or physically close to each other. If two people are in the same room the system assumes that the two people are talking to each other or are in a meeting. If such a situation is encountered, the system tries to find out, which documents are relevant. The documents "enter" the same room and upload themselves to the participants' client devices. Depending on the device, on the role of the person in the discussion, and the kind of meeting, the documents decide which view they choose. Hence, in some cases only a summarized version of a document is presented, whereas in a different situation a more detailed variant with attached references might be produced.

Thus, in this project, the active documents are aware of their content and of the context they are operating in. They make themselves proactively available to users and deliver what they determine to be the most appropriate representation.

6.4.3 University of Pittsburgh

A research group at the University of Pittsburgh describes an approach for the "fusion" of multimedia information by means of active documents advertising (e.g., [Chang and Znati 2001]). Fusion is defined as a "spatio-temporal integration of consistent as well as inconsistent information of different media and modalities from various sources" (from [Chang and Znati 2001]). This basically means that similar and in certain cases also dissimilar documents are logically connected with each other and forwarded to users.

An example: a person is working on a report about the rainforest in South America, and therefore the researcher might also be interested in articles that also deal with the same topic. In a conventional system, the user would have to employ a search engine or a digital library in order to find appropriate documents. The active document approach of the University of Pittsburgh, however, presents an architecture that facilitates this kind of information retrieval.

In this model, the user expresses the characteristics of a document in a set of metadata including, for instance, the document type, keywords, a document category, but also relationships such as "can be used as" or "definition of." Manual definition is preferred over automatic metadata extraction because it offers the user the possibility to put emphasis on certain aspects of the work. Based on the metadata the document *itself* starts actively searching the internet for documents with resembling features. Documents that are sufficiently similar are reported to the user.

The mechanism employed by these active documents works in two ways: on the one hand, the active documents search for other documents. These can either be active documents themselves or traditional documents. On the other hand, active documents also advertise themselves to other documents by propagating their metadata set. This is done by means of "adlets," light-weight application programs that are attached to a document and can perform a certain task.

The documents in this proposal are active in that they act autonomously. They work like special purpose search engines and, additionally, make themselves known to other documents. The main benefit is that users does not have to search for documents themselves. Thus, they can save time and focus on the actual work. A disadvantage is that some relevant documents might be unnoticed by adlets. (This problem, however, can also occur in traditional research without adlets.)

6.5 Conclusion

This chapter introduced Maurer's approach to active documents, a way to let users interact with documents. Some examples and aspects of an implementation were shown.

However, some other projects were also presented: Xerox PARC's active "placeless" documents, the active documents for a ubiquitous computing environment by the University of Stockholm, Sweden, and an agent-based approach to active documents by the University of Pittsburgh.

So far, only Maurer's approach has been incorporated in commercial products such as document management systems and e-learning environments. The other projects remained largely research prototypes.

The next chapter describes, how Maurer's model of active documents can be combined with other techniques in order to form a complete knowledge management system.

7 Extensions to Active Documents

7.1 Introduction

In modern knowledge management (KM) systems, seldom is only one single technology exploited. Usually, a number of techniques are combined in order to enable processes such as knowledge discovery or proactive dissemination. Active documents are one of these techniques and can play a key role in KM systems.

This chapter attempts to provide a "bigger picture" of active documents. It describes how they are used in a larger environment together with other components. First, an exemplary digital library is briefly explained, and several concepts that complement active documents are outlined. The second half of this chapter describes active documents in knowledge management environments and some characteristics of these systems.

7.2 Digital Libraries

Modern digital libraries are no longer passive document storage systems but include sophisticated functions that originate from knowledge management environments (e.g., [Maurer 2001; Hicks and Tochtermann 2001]). These functions are usually independent of each other. This means that active documents, for example, are not directly dependent on citation linking (see below).

Figure 1 depicts an exemplary digital library system. Each function of the system is depicted as separate module. The information the modules consume or generate is stored in the content and metadata archives of the library system.

It should be remarked that most digital libraries are implemented in a web-based environment. This is not a requirement, though.

The following sub-sections present three selected methods that complement the active documents functionality and are frequently used in digital library systems.

7.2.1 Annotations

Annotations give users the opportunity to add notes to documents or to certain sections of a document. Moreover, users are able to highlight regions within a document, just as they would be on a piece of paper.

In many digital libraries, annotations can be made on several access levels. This means that users can make private annotations that cannot be accessed by anyone else. Alternatively, annotations can be available to a certain group of users or can be publicly accessible. Group annotations can, for example, serve as the basis for the discussion of a document ([Maurer 2001]).

The concept of annotations is not restricted to textual content. Annotations can also be attached to multimedia objects such as images, sound files, or video clips. In fact, even annotations themselves do not have to be text-based but can be multimedia objects themselves. (The aspect of annotations in multimedia documents is addressed in chapter 8.)

From a technical perspective, annotations are closely related to the iconic approach in active documents (see section 6.2.3). First, users select a certain region in the document, and then they make an annotation. The system stores the document identifier and the exact location within the document together with the user's annotation.

7.2.2 Citation Linking

One of the most common features in digital libraries is citation linking (e.g., [Hitchcock et al. 2002]): instead of only mentioning the source of a quotation, a hyperlink to the original resource is inserted. Hence ideally, the user clicks on the hyperlink and has immediate access to the original paper – without having to search for it.

The main benefit is the facilitated and accelerated retrieval of references. A drawback of this approach is that links in a web-based environment are unidirectional. This means that if a document A references a document B, the latter will usually not be “aware” of A.

An attempt to solve this problem was made in [Garfield 1955]: an independent citation database keeps track of the citations of a large number of papers from different sources, diverse journals, etc. The main task of the database system is to organize and interconnect citations, and basically it maintains archives for the relations “refers to” and “cited by”. Therefore a citation database enables queries such as “which papers cited document A?” (See, for example, [Citeseer 2002; Lawrence et al. 1999] and [ISI 2003].)

It should be emphasized that citation databases are independent, i.e., they are not part of a digital library but can contain sources from a wide variety of heterogeneous, digital libraries.

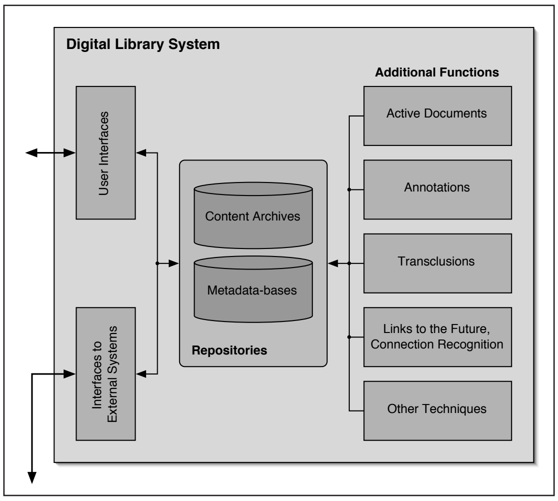


Figure 1: Exemplary architecture of a digital library system.

The idea of citation databases is taken one step further in what [Maurer 2001] calls “links to the future.” By quoting a previously published document B in a new document A, the author manually creates a reference $Ref_A(B)$ from A to B. In an automated procedure, the digital library system *itself* appends a reference $Ref_B(A)$ to document B. Thus, a “link to the future” from document B to document A is automatically generated. When researchers are reading B, they can easily see what the most recent contribution to the original paper is, and can access it immediately.

The drawback is that this approach needs to modify the original document (B, in the example above). This is usually only possible if both documents A and B are stored in the same library. Thus, it can be argued that “links to the future” are easily applicable only within homogenous environments. A system that implementes this feature is the Journal of Universal Computer Science (JUCS, [JUCS 2003]).

7.2.3 Transclusions

Transclusions, first mentioned by Ted Nelson in 1960, are one of the fundamental concepts behind hypertext ([Nelson 1981; Nelson 1996]). With transclusions, it is possible to extract a piece of information from one document and include it in a new document without duplicating the data. This is achieved by creating a reference to the original object (as opposed to copying and pasting it).

The main advantage is that the original context and metadata are preserved (e.g., [Krottmaier and Maurer 2001; Krottmaier 2002]). Traditionally, an author pastes a quotation along with the reference to the resource into a document. Hence, the original metadata including information such as keywords or an abstract is lost. Transclusions avoid this kind of problem.

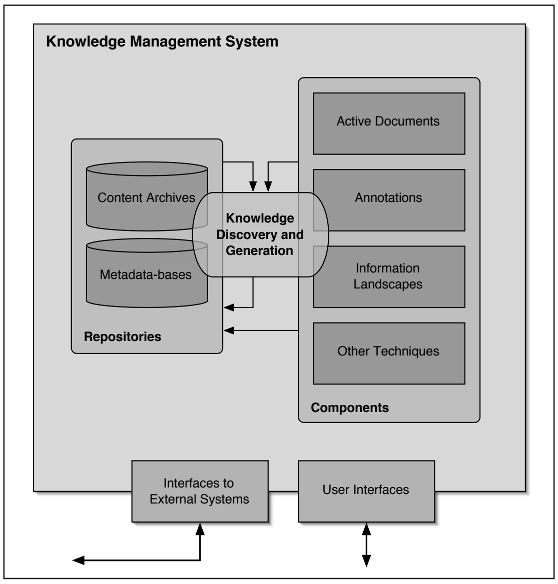


Figure 2: Exemplary architecture of a knowledge management system.

A disadvantage is that transclusions are difficult to implement. Therefore few digital library systems include this functionality.

7.3 Knowledge Management Systems as Extensions of Digital Libraries

Knowledge management systems (KM-systems) basically build on the functionality offered by digital libraries. However, the separate components play a more important part: active documents, for example, are not only an additional feature or a new function that can be utilized by users. They are also a source of data and can be an integral component of the entire environment. The active documents component works together with other modules, and the data collected by the modules is analysed and organized by the KM system (see [Maurer and Tochtermann 2002] and figure 2).

The difference between digital libraries and a KM system can also be seen in figures 1 and 2. In figure 1, the environment consists of several independent modules that store their data in the common content and metadata archives.

In contrast to this, the components in figure 2 are interconnected. The information provided by the modules is used together with the data that is retained in the archives of the system in order to discover and generate new knowledge. Both the information provided by the components and the newly generated knowledge are stored in the internal repositories of the system.

The following sub-sections present three selected technologies that are used in knowledge management systems. They can supplement active documents and also show how the data collected in active documents can be reused.

7.3.1 Personalization

Personalization in the context of KM means that a computer system "knows" the user. This is achieved by the KM system that retains a history of the user's actions. The information is used in order to predict future actions, to proactively provide documents and information, and to offer customized objects or services. (See also section 7.3.2 below.)

In many KM-systems, personalized services are provided through personalized digital libraries (see [Hicks 2001; Westbomke et al. 2002]). One of the techniques that is frequently employed in this domain is case-based reasoning (CBR, see chapter 3). This means that the user's actions are stored as experiences in a case-base. When the system comes upon a situation that is similar to a previously encountered one, it can offer users appropriate, personalized services to support them.

In this approach, active documents play an important role because the questions asked by users are a valuable source of information; they can be stored in the case-base. Also feedback to the answers provided by the active documents component can be used in order to enhance the quality and accuracy of the provided services.

7.3.2 Proactive Dissemination

The Maurer-Tochtermann-Model of knowledge management introduced in section 1.6 describes several different types of input and output for KM systems (see figure 3 in chapter 1). The model also includes the implicit "retrieval" of information, i.e., the system offers information and services to users without their explicitly requesting them. This process is commonly referred to as *proactive dissemination.* A practical example for proactive dissemination are lesson learned systems (see section 2.3 and [Abecker et al. 1998]).

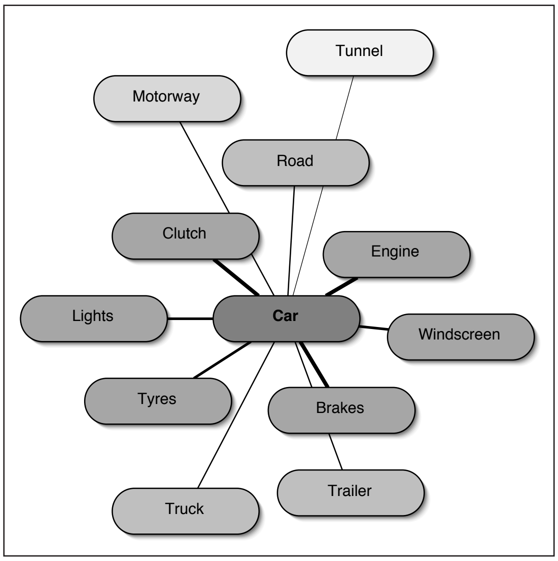


Figure 3: A simple information landscape for the term "car."

The KM system needs to collect data about the user in able to determine which information is to be disseminated to the user. This data can, for instance, stem from users' annotations or from questions that were asked in active documents. In this case, the user's input to the KM system is implicit because the KM system *itself* extracts and filters the relevant information. The implicit user input is also indicated in figure 2 where the KM core collects data from several components and stores it in the repositories of the system.

7.3.3 Information Landscapes

Information landscapes are a technology that is well-suited to graphically depict relations among documents and (multimedia) objects (e.g., [Maurer 2003; Mülner 2001]). The generation of information landscapes is done in two steps: first, the connections between two documents or objects have to be determined, and subsequently the most significant connections are represented in a map. The detection of connections includes several techniques (see [Dallermassl and Helic 2003]):

- word and pattern similarities,

• manual and automatic classification, and
• link weights.

In the first stage, text documents are compared for similar words, images for similar textures, and other media objects for similar patterns. The classfications of the objects and documents are also analysed: if two objects are in the same or a similar category, a connection exists between them.

In order to emphasize more significant documents, link weights are calculated and considered: a document that is connected to a great number of other objects has a relatively high weight and is therefore presented close to the center of an information landscape. In a similar way, the link weights of connections to the object in the center are calculated, and objects with a higher link weight are depicted closer to the center of the information landscape.

Figure 3 shows an information landscape for the term "car." The term itself is the depicted in the center of the map. Objects with more relevant relations to a car such as the engine or the brakes are closer to the center, whereas other terms such as "truck" or "tunnel" are further away from the center.

Thus, an information landscape can highlight important objects and show which other objects are related to it. Semantic object maps, a similar technology mainly used for the description of the structure of video content, are briefly described in section 11.4.5.

7.4 Conclusion

This chapter gave a overview of the two major areas in which active documents are employed: digital libraries and knowledge management systems. In digital libraries, the basic active documents functionality is complemented with a number of other features such as annotations and links to the future in order to provide a set tools that support the user.

Knowledge management systems build on the technologies introduced in digital libraries and extend their basic functionality to provide more sophisticated functions such as personalization or proactive dissemination. In KM systems, active documents are not seen as a separate component that offers a distinct function but as an essential part of a larger environment. Active documents are not only a tool that provides a certain function, but also a source of data, and information that can be related to other information in the KM system.

Active Multimedia Documents

8

8.1 Introduction

Although the active documents paradigm described in chapter 6 is not limited to text-based representations, its usage in media-rich domains has not yet been discussed. Therefore this chapter details how active document functionality can be employed with multimedia documents such as images, video clips, and sound.

First, a number of supplementary and alternative input methods for use with multimedia data are introduced. The subsequent description of active multimedia documents makes use of these techniques. Finally, some aspects concerning the storage and organization of active multimedia documents are addressed. Based on the findings in this chapter, chapter 9 presents an exemplary application of active multimedia documents in the medical domain.

8.2 Supplementary Input Methods

The following few sub-sections describe a selection of input methods that can be used in conjunction with active documents. These alternative input methods focus on ways to supplement the classic paradigm of asking questions to documents. They are especially well suited for multimedia documents.

8.2.1 Region Selection

Region selection is a simple yet powerful technique. It lets the user select a piece of information in any kind of media object such as one paragraph of text, an area of an image, two seconds of music, etc. Region selection can be applied for text, visual content (e.g., images or drawings), audio material (sound), audio-visual information (video clips), and similar media.

Advantages of this technique are that it is comparatively easy to implement, it is expressive, relatively unambiguous, and it is flexible in that it can be used for many different kinds of media independent of the problem domain. Moreover, users do not have to have special skills to make use of this technology. A disadvantage is that the method is not very sophisticated, and therefore more complex attributes or relations cannot be articulated.

8.2.2 Texture and Pattern Descriptors

Texture descriptors can be used to describe visual characteristics of images and video content (e.g., [Tamura et al. 1974; Photobook 1994]). The more general concept is a pattern description that can be also used to define patterns in other media such as sound and music. Texture and pattern descriptors are often combined with query by example functionality (see below), i.e., the user can select a texture or a pattern from a predefined set of examples.

In active document systems, texture and pattern descriptors can be used to attach annotations and to ask questions that correspond directly to pattern-related characteristics. A benefit is that pattern descriptors enable users to specify properties that are otherwise rather difficult to describe. Drawbacks are that the technique can be ambiguous, and that a certain experience on the part of the user is required.

8.2.3 Query by Example

The query by example (QBE) paradigm was introduced in [Zloof 1975]. Instead of making users explain features verbally, QBE lets them give an example for a relationship or a certain characteristic they wish to describe. The examples can stem from a precompiled list of examples or can be freely chosen and provided by the user.

The original idea of QBE was implemented in a text-based database system. However, recent multimedia storage and retrieval systems employ this concept also for images, videos, sound, and similar data (e.g., [Chang et al. 1997]). Active document systems can utilize this technology to allow users to describe documents or features thereof that are *just like* a piece of information they have already found.

A benefit of QBE is that it works on the perceptual level, and therefore it is a natural way of input for users. It enables the use of common relations such as "is similar to." However, this method is sometimes imprecise because the user might focus on a detail in the provided example, viz., a detail that cannot be perceived by the QBE component. Furthermore QBE systems are often domain specific, rather difficult to implement, and for many problem domains development is not beyond prototype status.

8.2.4 Motion Sketch

With motion sketch technology, the user has the capability to specify the motion that an object performs within a certain period of time (see [Nakamura and Asada 1995]). This information can, for instance, be used to find similar objects or to describe elements of a video document. An example is illustrated in figure 1: first, the user vaguely outlines the shape of the object (a circle, in this case).

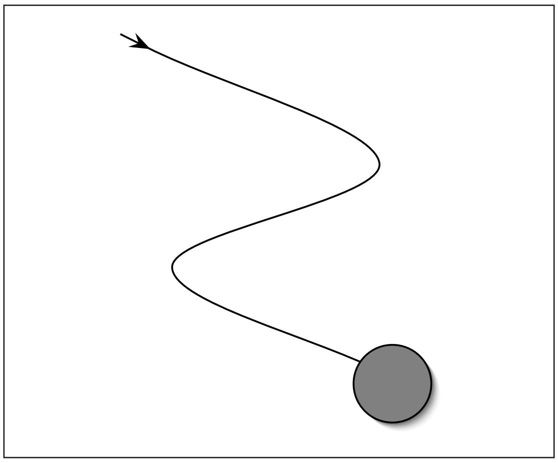


Figure 1: Motion sketch as input for an active documents query. This motion sketch could, for example, stand for a slalom skier or for the "flight path" of a butterfly.

Then, the motion of the object is specified by drawing a vector. Additionally, the duration of the sketched motion can be defined by the user in order to make the definition more accurate.

In active document systems, this method can be employed to express the motion of elements in a video clip. Based on the motion sketch, the user can ask a precise question or make annotations that correspond to the motion of the object.

Motion sketches are a very powerful tool for video content because they are very expressive and meet the users' demand to describe and query motion pictures. A drawback is, though, that this technique is difficult to implement. Moreover, users need to have a certain degree of experience to define motion sketches efficiently.

8.2.5 Speech, Sound and Music Input

Although sound and music input as well as input in natural speech are very important, these techniques are not detailed in a separate sub-section. Sound and music input are partly covered by texture and pattern descriptors as well as query by example methods (see above).

Speech input, on the other hand, is increasingly implemented on the operating system level (e.g., [SRecMgr 1997; SRecMgr 2003; FreeSpeech 2003] see also [Padmanabhan et al. 2001]). This means that speech recognition is provided by the operating system, and that it is not only exclusively available to one particular application program, but to all software components running in this environment. Hence, a separate discussion of speech input is not necessary at this point.

8.3 Multimedia Documents

The concept of active documents can basically be applied to any document type. Therefore, this section briefly explains a variety of multimedia document types, and their use in connection with active document features is addressed.

The methods introduced in the following sub-sections can make active document queries more precise and more efficient, and enable users to make annotations more easily. An application of active documents in e-learning environments that uses some of described features is introduced in the next chapter.

8.3.1 Drawings, Vector Graphics

Drawings and vector graphics are used very often in the technical domain and when abstract concepts are graphically depicted. Examples are CAD drawings including plans of buildings, cars and machinery in general; both two- and three-dimensional models of objects such as molecules in chemistry; flowchart diagrams and organizational charts.

A common property of most vector graphic formats is that every vector and every other object is stored in way that it can be addressed independently. This means that even after saving a document, a line can still be selected as a line, and its length or position can be modified.

The same is also true of some formats that are employed to describe three-dimensional model or "virtual reality" scenes. The commonly used VRML format, for instance, stores every object separately in the file and it is possible to modify every object separately (see [VRML97 2002]).

This characteristic can be made use of in an active documents environment. When users want to ask questions, they can select a certain object first, and the question they ask corresponds to the selected element. This means that users can select a certain detail in a plan of a house, for example, and ask a question or add a comment that is related to this object. An example of a selected object in a vector graphic is shown in figure 2.

8.3.2 Photos and Images

Photos and rasterized images are usually produced by conventional photo cameras, specialized cameras such as infrared cameras, and other imaging devices such as ultrasound and x-ray detectors or radar units. As diverse areas as medical imaging, satellite imaging, microscopy, (print) publishing, or the WWW make use of rasterized images.

Also some technologies for the description of three-dimensional models and virtual reality scenes such as Apple's Quicktime VR utilize rasterized information (e.g., [QTVR 2002]). Images are stored and represented in a particular way that makes it possible to display them as three-dimensional objects.

For rasterized images it is usually not possible to select separate objects, only pixels or regions of pixels can be addressed. Thus, a straightforward approach for images and photos is selection based on regions (see section 8.2.1 above, see also figure 4). When users want to ask a question, they can mark a certain (spatial) region of the image first, and their question or annotation refers to the selection.

Further techniques that are suitable for images are texture-based queries (see section 8.2.2 above and [Tamura et al. 1978; Photobook 1994]) and query-by-example functions (see section 8.2.3 above and [Chang et al. 1997; Photobook 1994]).

8.3.3 Video Content

Most video content is represented as a series of frames, where each frame is a rasterized image. Therefore basically all methods that can be utilized with images and photos are also applicable with video content. The selection of a spatial region, for example, is depicted in figure 4. In addition to this, three further region selection mechanisms can be identified:

- temporal selection,
- spatio-temporal selection; and
- spatio-temporal selection with motion sketches.

When using a temporal selection, the user specifies a certain period of time on the time line of a video clip in order to denote that the question or annotation refers to this temporal region (see figure 3). The combination of a spatial and a temporal selection leads to a spatio-temporal selection. This means that a fixed area in one frame of a video clip is selected also in a number of consecutive other frames. This method is useful, for example, when the camera films a static object for a period of time. The object as such does not move but some of its attributes such as the colour might be altered over time.

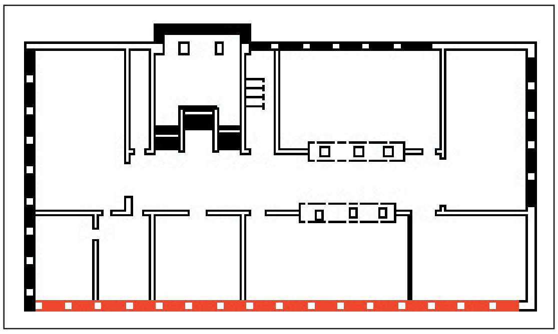


Figure 2: A part of the plan of the Bauhaus in Dessen, Germany, stored as vector graphic. One object (a wall) was selected by the user and is highlighted. Image source: [Kulper 2002].

The third technique is a spatio-temporal selection with motion sketches. In this paradigm the user can define how a particular region moves during a certain period of time. First, the user selects a spatial region in the video clip. Subsequently, the period of time has to be determined (temporal selection), and the movement of the spatial region has to be described.

An example is illustrated in figure 5: the user selected the ship as spatial region, defined a certain interval on the time line of the video player, and expressed the movement of the selected object with a simple motion sketch. The query in that example relates to the ship that moves from the start position to the end position within a given time.

8.3.4 Sound and Music

A number of different approaches for describing and querying sound and musical data exist, and several systems

Figure 3: Video clip with a temporal selection.

Figure 4: Video clip with a spatial selection.

Figure 5: Video clip with a spatio-temporal selection and a user-defined motion sketch.

for music retrieval, automatic sound detection and sound recognition are available (e.g., [Polyphonic 2003; Lemstrom and Perttu 2000; Defaux et al. 2000; STC 2002] and [Sphinx 2002; Casey 2001; Omras 2000]).

For the description and selection of music and sound in active document systems, two approaches seem reasonable because they are relatively easy to use: region selection and query by example. Region selection means that the user marks a temporal region of a audio document (conform with figure 3). When the query-by-example paradigm is employed, users have to provide an example of a music or sound document. Additionally they can specify which features of the given example are most relevant: volume, pitch, timbre, etc.

An example for the combination of sound and active documents is a library of digitized music, where users can asks questions referring to a song or can make annotations to a part of a song.

8.3.5 Compound Documents

Compound documents consist of several, largely independent elements of different media types. Examples are multimedia presentations that may contain text, images, sound, and video data, or HTML pages that can include images or video clips.

Since compound documents are basically made up of a set of basic documents, the techniques introduced above can be utilized. First, users have to select a particular object within the compound document, and subsequently, they can apply methods such as region selection or motion sketches.

8.4 Storage Aspects

Aspects of storing active documents information are important in larger knowledge management envioronments because they determine the reusability and efficiency of the system. Therefore the storage concepts described in this section attempt to follow established standards such as MPEG-7 or elements of MPEG-21.

8.4.1 Object Selection and Description

Before objects and related question-answer pairs can be stored they have to be identified and described appropriately. The storage of active multimedia documents data typically includes the following steps:

- unique identification of the media object;

- description of a region, selection of an example (for QBE), specification of a motion sketch, etc.;
- generation of a question-answer pair or an annotation;
- storage of the question-answer pair or the annotation with a connection to the selected region.

Uniform resource identifiers (URIs) for the identification of media objects can rely on standards such as the Digital Object Identifier initiative (DOI), ISAN for audio-visual content, ISRC for music, or ISBN for books. For details see section 5.3.5 on the MPEG-21 digital item identification.

In the next step, the select region has to be specified. For images this can, for instance, be the (x,y) coordinates, and for videos the time code or the number of the frame can be stored. When the query-by-example concept is applied, the provided example together with the relevant features and other parameters can be stored.

When the annotations and questions-answer pairs are stored, the URI together with the specified region can function as primary key for a database entry.

8.4.2 Storage

There are several ways to retain the collected data. One distinct approach is to use a metadata standard together with a database solution.

The selection of certain regions within objects, and annotations as well as question-answer tuples that are attached to these objects can be seen as metadata. Therefore an implementation of active multimedia documents could, for instance, use MPEG-7 for the description and storage of this information. The generated MPEG-7 descriptors are stored in a database system such as eXist, a general purpose XML database system (see [Meier 2002; Meier 2003]).

The advantage of this approach is that the data collected through the active documents functions can contain valuable metadata. By storing the information in a standardized format such as MPEG-7 greatly facilitates the reuse. The data originating from active documents could, for example, be employed in order to enhance the querying of multimedia databases, etc.

8.5 Conclusion

This chapter introduced several ways to apply active document features to a wide range of multimedia documents. Some supplementary input methods were presented, and the use of active multimedia documents was shown.

Based on the techniques and ideas outlined above, the next chapter proposes a concept for active documents in medical education, and Part 3 of this thesis makes use of active multimedia documents in video broadcasting environments.

Active Documents in Medical Education

9

9.1 Introduction

This chapter presents the idea of a system that uses the functionality of active documents to facilitate and enhance the learning process in medical education. It introduces ADIME, a system proposal for **A**ctive **D**ocuments **i**n **M**edical **E**ducation.

First, the current situation of e-Learning in medical education is outlined. Subsequently, the need for active documents in this domain and the basic idea of a new system are explained. Based on these findings, ADIME is introduced, and the design of a prototype that implements the proposed functionality is outlined together with the underlying system architecture.

9.2 The Current Situation of e-Learning in Medical Education

e-Learning in medical education has a long tradition, and a very large number of competing products exist at the moment (e.g., [Langkafel 2002a,b; Wagner 2003; Wagner and Hansen 2002]). A rather brief survey of e-learning initiatives in Germany yielded the following results: [AGMA 2003], for example, lists more than 200 different German programmes available both on CD-ROM and on the WWW, and [Hierl et al. 2003] notes about two dozens of "new", computer-supported projects in education in medical education.

However, many of these projects are very similar and introduce hardly any new features or technologies. They "re-invent the wheel." Thus, most e-learning systems that are currently employed in medical education are mere information- or document management systems. They lack characteristics of knowledge management (e.g., [Karsten and Neumann 2002]).

Education in the medical domain does not only rely on text-based material but makes, for instance, intensive use of images. Examples are anatomy, histology, the science of organs and tissue (e.g., [Vollrath 2003]), and pathology, the science of diseases (e.g., [Herbst and Huebner 2002]). These disciplines utilize images and recently also animated, three-dimensional models. Therefore, there are also several systems specializing in multimedia content with medical relevance. These are, for example, image databases and video clip archives (e.g., [Medianovo 2003]). However, these environments are fairly static and passive and focus mainly on:

- automated image processing: enhance the quality of images and video clips;
- indexing and retrieval: store information and media in a way that it can easily be found and retrieved;
- so-called "intelligent search functions": full-text search and querying for metadata that can be attached to images, video clips, etc.;
- ubiquitous access: retrieve data on a PC at home, with a computer in the lecture hall of a university, or a special terminal in an emergency room of a hospital;
- semi-automatic generation of training courses and textbooks from existing content in the media databases.

So although e-learning systems in medical environments utilize multimedia content of diverse types, the use is mostly quite conventional, and knowledge management has not played a big role in this domain so far. The media documents are merely passive, and the user can usually not interact or communicate with these objects.

One of the few applications of knowledge management in medical information systems is Infomed-Austria, a medical meta server in German language (e.g., [Maurer and Guetl 2000]). However, this system is not intended for e-learning. Moreover its focus is rather on text-based information (that can be supplemented with images).

9.3 The Basic Idea

During the preparation phase for anatomy or pathology, for instance, medical students are dealing with a big amount of image material. Sometimes it is unclear, though, what a picture is supposed to show, which region of a picture is important, or if a detail of an image indicates a normal or a pathological condition. Examples include untypical shape or colour of tissue, or unusual blood flow in veins. Therefore students occasionally have questions such as *"Is this condition pathological?"* or *"Is this detail important?"*

Conventional e-learning environments do not offer facilities to ask suchlike questions, and usually it is difficult to find appropriate answers within such a system. Active documents, on the other hand, offer this functionality, have successfully been implemented, but are used in largely text-based systems (see chapter 6).

Hence, the basic idea is to apply active document features to multimedia documents in medical e-learning environments. Users and tutors are also provided with a method to add (private or public) annotations to textual, image, and video content. Thus, this concept introduces features of knowledge management in medical e-learning systems. Furthermore, it can be a valuable tool in medical education in general.

The focus of this chapter is on the medical domain because, due to the large amount of visual information, it is well suited for such an approach. However, the basic methodology can be applied to almost any information and knowledge management system that handles multimedia data. Examples for other domains are images of circuitry in electrical engineering, plans of buildings and corresponding three-dimensional models in architecture, two- or three-dimensional models of molecules in chemistry, etc.

9.4 ADIME

The above description of the basic idea leads to ADIME, a proposed system for **A**ctive **D**ocuments **i**n **M**edical Education. Based on active documents (see chapter 6) and active multimedia documents introduced in chapter 8, a novel application for more interactive e-learning is introduced.

9.4.1 ADIME User Interface

The ADIME application program typically can implement the functionality of a video player or an image viewer or both. Figures 2-4 illustrate an exemplary user-interface for an image viewer. Initially, the user has a picture displayed. This is shown in figure 2. The basic user interface resembles a traditional image viewer. The proposed, new functions are available through the two buttons "Show Annotations" and "Ask New Question."

The user looks at the picture and has a question. Therefore the user selects the "Show Annotations" function to find out if a tutor or other users have already attached annotations to this image (see figure 3). In a "drawer" on the right-hand side of the window, all annotations are displayed. Additionally, previously asked questions and the related answers are also presented in this area. (In this example, no questions have been asked so far.)

The only annotation that is available refers to a certain region of the image that is highlighted. Annotations and existing questions rather displayed on the user's request and are not communicated pro-actively. If annotations and highlighted regions were displayed in the initial view (figure 2), the users' attention would probably be attracted too much to the highlighted areas, and they would actually not discover the essence of an image themselves.

This can be compared with students doing their homework in maths: usually they get an assignment, and have to solve the tasks themselves. If they also had the solutions the method to solve the problem would be obvious, and the homework would not the desired effect (e.g., to learn how to solve problems or to find a problem-solving strategy).

When users cannot find an answer to their questions by browsing the annotations and previous enquiries, they can ask new questions. Pressing the "Ask New Question" button causes a new drawer on the right-hand side to open. This drawer contains tools for selecting regions in the image and an area for asking the actual question (see figure 4).

Usually, the user highlights the region which the question refers to, first. This can be done with a number of different tools including a pencil and a rectangular and an elliptical marquee. If the user does not define a specific area, the question corresponds to the entire picture.

Then, the user asks the question, typically by typing in text. In figure 4, some frequently asked questions are predefined. Alternatively, the user can freely enter an arbitrary question in natural language.

The process of asking a question is not strictly limited to typing in text on a keyboard. More sophisticated techniques can be employed. An example is to provide a separate area in which users can draw their own sketches. The corresponding question might be *"Should the highlighted region in the picture have the features in my drawing?"* Of course, not all techniques are reasonable in all problem domains, and some might not even be applicable. Thus, the input methods that are to be implemented have to be considered individually, based on the problem domain and the users' needs and demands.

9.4.2 Technical Perspective

After having described the basic idea and the functionality of ADIME, some technical aspects have to be addressed. It should be noted that ADIME is a concept rather than a concrete software system. Therefore it can be implemented in a number of different ways and can be used together with a wide variety of existing systems.

Just as active documents are an add-on to conventional document management systems, digital libraries, etc., ADIME can be understood as an add-on to traditional e-learning environments (see section 9.5.3 below). The technical perspective of the proposed application and the architecture of a system implementing those functions are discussed in the next section.

9.5 Implementational Aspects

This section outlines how ADIME can be implemented in a networked computer environment. Apart from the design of a system architecture, also some aspects related to the chosen user interface and issues concerning active document features are discussed.

9.5.1 User Interface

The user interfaces in figures 2-4 show a specialized application program that will provide the proposed functionality. Typically, such an application is used on a desktop computer. This is, however, only one particular instance. The same functions can also be implemented with a web-interface: the HTTP server displays the images with the corresponding text. If the user wants to select a region of this image and ask a question, a Java applet can be loaded, and the further interaction is carried out by the applet.

Other user interfaces are feasible as well: an application program for mobile devices such as handheld computers with wireless internet access, for example, or an ADIME software package for use with a terminal computer in a lecture hall or a library.

9.5.2 Details of Active Documents Features

As for any active documents application, the stated convergence criterion of 500 to 1,000 users per document has to be met (see section 6.2.1). When ADIME is used as courseware for a lecture in an e-learning environment, this requirement might not be met within one semester. However, the contents of the documents are rather static (the human anatomy, for example, does not change), and the number of users grows over time. Thus, the convergence criterion of some 500 to 1,000 users per document can be met within a few semesters.

For the implementation of the active document features in this application, the iconic approach is chosen (see section 6.2.3). This means that users first select a certain region on the screen and ask a question that refers to this section. For still images the region is spatial, for video clips it can be spatial, temporal, or spatio-temporal.

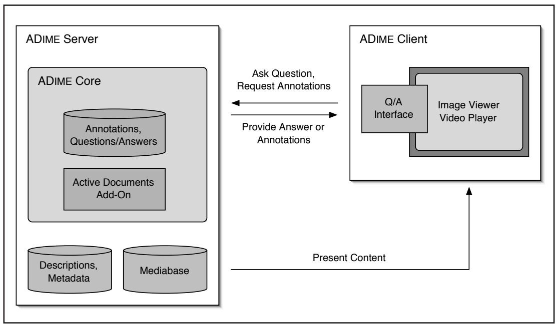


Figure 1: Design of a system architecture for ADIME consisting of an ADIME server with the ADIME core and an ADIME client running an image video or a video player (or both).

The algorithm for checking the congruence of two regions can be quite simple: if two spatial areas overlap to a given percentage, they are marked as matching. The same idea can also be applied for temporal regions.

In addition to selecting regions, more sophisticated techniques can be employed. Motion sketches, for instance, can be used to ask questions that are related to the movement of a particular object in a video clip (see section 8.3.3 and [Chang et al. 1997]). Examples are the motion of a blood clot in a vein or the spread of an injected colourant in an organ.

Further features that can be implemented include the application of Tamura textures (see [Tamura et al. 1978]), functionality from query-by-exampe systems, etc.

9.5.3 System Architecture

Largely independent of the chosen user interface technology, the proposed system describes basically a client-server model. The design of a system architecture for ADIME in a networked environment is illustrated in figure 1.

The system consists of a server that stores images and video clips as well as corresponding explanations and

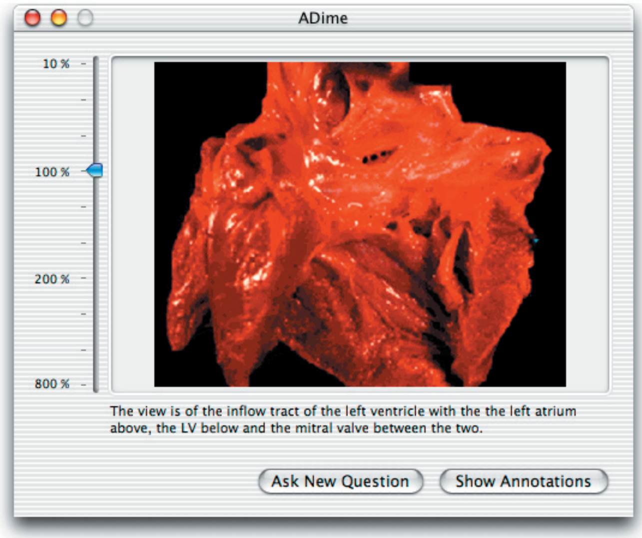


Figure 2 (top): A prototype for the ADIME image viewer. The central user interface resembles a conventional image viewer with an area for displaying the image, a small text area for some basic metadata, and a zoom function.

The new functionality of the proposed application is accessible through the two buttons "Show Annotations" and "Ask New Questions".

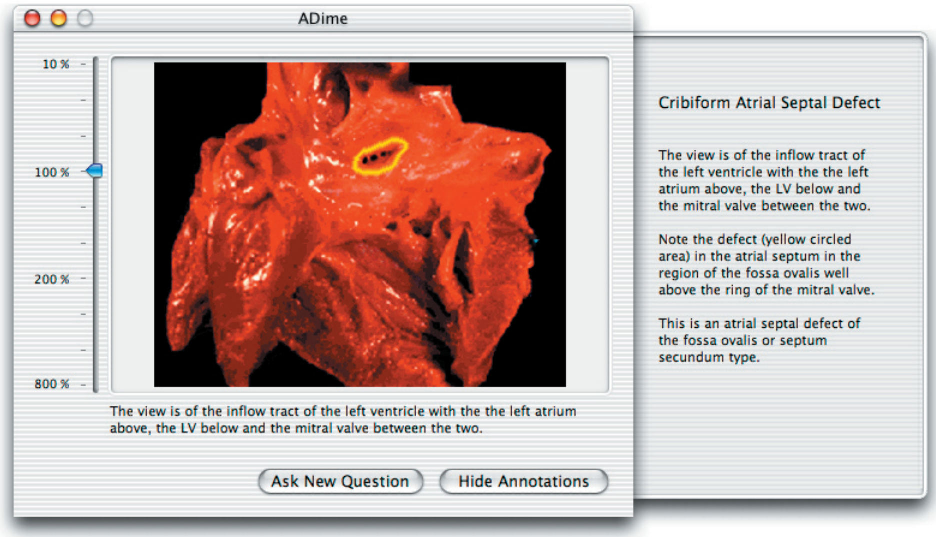


Figure 3 (middle): The user has pressed the "Show Annotations" button in order to view annotations that have been made either by other users or a tutor. Moreover, questions that have been asked by other users are displayed, if any are available.

In this example, the tutor has made an annotation that refers to a certain region in the picture. The region is highlighted, and the "drawer" on the right-hand side of the window contains an explanation. Users have not asked any questions concerning this picture, yet.

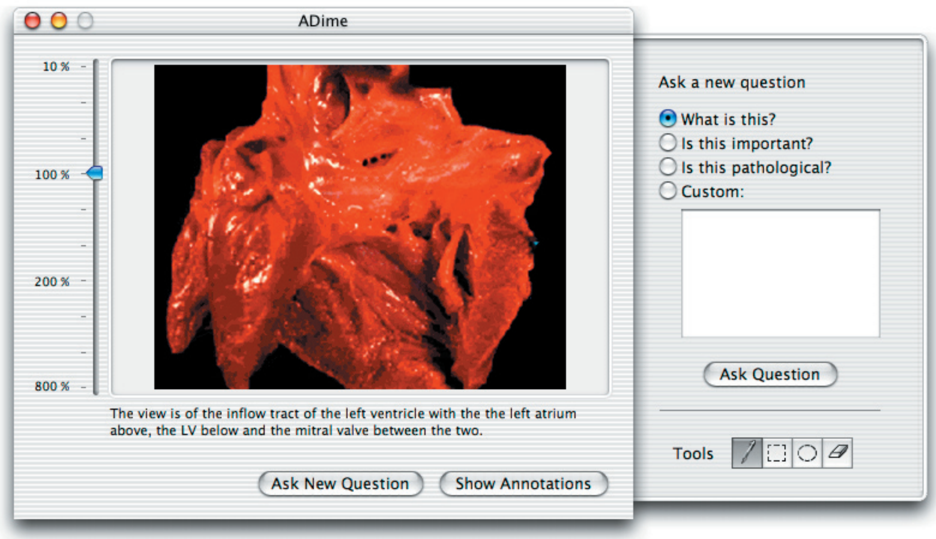


Figure 4 (bottom): The user has chosen to ask question. First, with one of the tools (pencil, rectangular or elliptic marquee) a region of the image can be selected. If the user highlights a certain region, the question refers to it. Otherwise the question refers to the entire image.

Subsequently, the user can ask a question. Frequently asked questions such as *"Is this pathological?"* are available as predefined options. Alternatively, the user can freely specify a question by entering it in the provided text field.

The medical photograph of the heart is taken from [Hasson 2002].

metadata. The server's system core contains the active documents add-on (see section 6.2.2) and a repository for the collected question-answer pairs. The ADIME client includes an image viewer or a video player (or both). Moreover, it contains a user interface element that lets the user ask questions and view annotations (denoted as "Q/A Interface" in figure 1).

The chronology of a characteristic client-server communication is as follows: first, an image or a video clip from the content database of the server is presented by the client-side application program. When the user wants to retrieve annotations or wants to ask a question, a request is transmitted to the active documents add-on of the server-side ADIME core. The request is processed by the active documents component, and subsequently, the related response is sent back to the ADIME client and presented to the user.

9.6 Conclusion

Most state-of-the-art e-Learning systems in medical education are rather static information management systems that include hardly any knowledge management features. ADIME, a proposed system for active documents in medical education, attempts to solve these shortcomings of conventional systems by introducing functions known from active documents.

With ADIME, users can be provided with a powerful tool that lets them find answers to content-related questions and allows them to retrieve annotations. This makes existing e-learning systems more active, facilitates knowledge transfer, and enables knowledge management.

Since the described architecture is an add-on to existing systems, it is flexible and can be implemented in a number of ways including standalone application programs and integrated web-based software components. Thus, ADIME offers reliable and effective technology for e-learning systems in the field of medical education.

Part 3

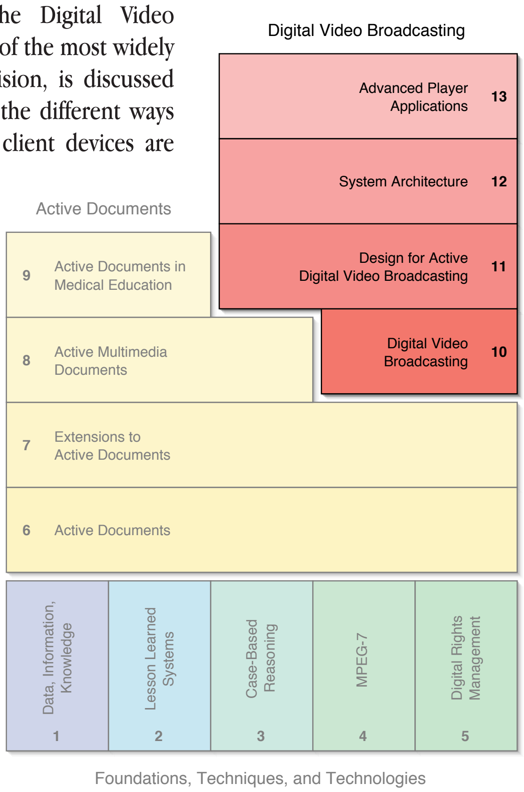
Active Digital Video Broadcasting

Active Digital Video Broadcasting

The third and last part of this thesis discusses how advanced features for digital video broadcasting can make television and other video dissemination technologies more interactive.

Chapter 10 establishes the necessary foundations for later chapters. It describes standards and technologies that are commonly used in digital television and digital video broadcasting environments. As an introduction to the topic, the role of MPEG-2 in digital television environments is outlined. The Digital Video Broadcasting specification, one of the most widely used standards in digital television, is discussed in greater detail. In particular, the different ways of transmitting digital data to client devices are explained. Furthermore the Multimedia Home Platform, a framework for developing software for consumer devices on top of the Digital Video Broadcasting standard, is described.

The next chapter introduces Vivid, a design for active digital video broadcasting. The fundamental concept is depicted together with the characteristics of such a system and its relation to active documents. A large part of this chapter presents several usage scenarios of the technology: movies, documentaries and newscasts, sports, children's programs, and music television.



The subsequent section focuses on more technical aspects, namely the different kinds of information and their role in the system. A few examples of how the stored information can be presented to the user are given. Ultimately, some economic aspects and a possible, simple business model are delineated.

From a technical perspective, the architecture of the proposed system can be seen as a client-server approach: the TV channel or video network provider is the server, and the consumer devices are the clients.

Chapter 12 presents the server-side component of Vivid. First, the design goals and a generic system are presented. After describing the basic system architecture, the key components of the generic system are detailed: the core system, the input and output modules.

Two particular models are based on the generic approach: an MPEG-4-based and an MPEG-2-based architecture. The remaining sections of the chapter describe the essential components of these two systems.

The subsequent chapter gives an overview of Vivid player applications. This is the software that runs on consumer devices and is able to decode and present the data required and delivered by the entire system. Thus, chapter 13 shows the client-side perspective of Vivid.

As an introduction, the architecture of the client-side player application is outlined, and some design goals are given. Moreover, the underlying technologies are addressed for different client platforms:

networked computers, mobile computing devices, and digital television sets.

Subsequently, two instances of Vivid player applications are demonstrated: a standardized and an extensible player. The first one is an application program with a limited set of functions that focuses on restricted hardware capabilities and rather inexperienced users. The latter variant is more sophisticated, offers extensibility and therefore potentially more functionality. Its focus is on professional users and more powerful hardware platforms with internet connections.

To complement the topic of digital video broadcasting, Appendix A contains an essay about the convergence of digital television and the internet.

Digital Video Broadcasting

10

10.1 Introduction

This chapter presents the elementary concepts of digital video broadcasting and demonstrates its underlying technologies and standards.

As an introduction, the field of digital television (DTV) is surveyed, and plans for the launch of DTV networks are outlined. Detailing the Digital Video Broadcasting Standard (DVB) leads to the Multimedia Home Platform (MHP), a framework for developing applications for DTV. The availability of such a technology has been expected for a long time by many vendors and is essential for the ideas described in the following chapters.

Finally, the architecture of an interesting product that could possibly be used as a digital video broadcasting consumer device is illustrated.

10.2 Digital Television

Most current television broadcasts including terrestrial, cable, and satellite television are based on analog technology. Widespread standards are PAL (Phase Alternate Line) and NTSC (National Television Standards Committee).

Although the vast majority of television equipment is still conventional analog technology, the television of the future is digital. This has several benefits for users: better image quality due to a much higher image resolution, new applications, interactivity, and compatibility with computers and computer networks are only some of them (e.g., [Wipro 2001] and [Lugmayr et al. 2002]).

Two more advanced standards for digital television are ATSC (Advanced Television Systems Committee, engineered and used in the USA) and DVB (Digital Video Broadcasting, developed in Europe). The latter is described in detail in section 10.3.

10.2.1 MPEG-2 in Digital Television

The underlying technology of all currently available DTV systems is MPEG-2. (The family of MPEG standards and MPEG-2 are briefly introduced in section 4.2.2.) It offers support for multiple audio and video streams at different levels of quality. One of the key features is that separate data and interaction channels are available.

Aside from audio and video, MPEG-2 distinguishes two different kinds of data: service information (SI) and private data. Service information contains data about the audio, video, or data streams transmitted with MPEG-2. Private data, on the other hand, can carry data that was requested (explicitly) by a user or data that is transported (implicitly) by the system to one or more user(s).

Each transmitted MPEG-2 stream contains several so-called *elementary streams* (ES), where one ES can only contain a single type of information such as (either) audio or video. Typically, a TV program consists of one video ES, at least one audio ES, and several other ESs for control data, subtitles, etc. Both audio and video data is organized in *access units,* where each unit is a characteristic portion such as a frame (see [Fairhurst 2001a-c] and [Morris 2002a]).

The MPEG-2 compressor processes the ES and produces a *packetized elementary stream* (PES) that is made up of packets of fixed or variable length. Each packet includes identifiers in its header that make it possible to determine what kind of packet it is and which ES it belongs to. In the next step the multiplexer prepares the PESs for transmission via program streams or transport streams.

Program streams are generally used in environments with low error rates such as DVDs. In contrast to this, transport streams are better suited for transmission systems in which loss or corruption can potentially occur. MPEG transport streams are therefore used in DTV. They are basically independent from the underlying transport media and can be used with a broad variety of different networks (see section 10.3.2 and table 1).

A transport stream can be a single TV program, for example, with one video channel and two audio channels. This is called a single program transport stream. Usually one single program transport stream does not utilize all of the available bandwidth (e.g., one satellite transponder), and therefore a part of the bandwidth would be wasted. This is the reason why in most cases several independent TV programs are packed into one transport stream to form a multiple programme transport stream.

In this case program specific information (PSI) has to be added to the transport stream to be able to coordinate and correctly de-multiplex the different channels at the receiver.

10.2.2 Requirements for Digital Television

One of the biggest challenges in the migration from traditional analog television to digital systems is to overcome the incompatibility of existing equipment with the new standards. Conventional TV sets cannot be used with the new technologies, but consumers will often *want* to reuse them.

Two approaches can be identified to resolve this issue: integrated digital TV sets and set-top boxes. A digital TV set is a complete new device that only has the shape and the CRT (or the flat screen) in common with the traditional appliance. The components that receive and decode the TV signal are completely different.

The receiver unit works in a similar way to that of the receiver of a digital cell-phone or a network interface does. The decoder, on the other hand, is a complex microchip capable of decoding and processing the incoming MPEG-2 stream. In a modular design the decoded data is then passed on to other elements such an audio component, a display component or an application processing component.

It is important to note that in this approach the display component is connected to the CRT or LCD, and the audio component is directly connected to the speakers. Hence, all modules are adjusted to each other, and the audio and video data can be used in the best quality. This is illustrated in figure 1.

In contrast to that, a set-top box (STB) is a kind of an addition to a conventional TV set, just like an analog satellite receiver or a VCR. All the logic for receiving and decoding the digital signal and decoding it lies within the STB. It decodes and processes the digital signal, and finally converts it to an analog signal so that the conventional TV set can use it.

This technique seems to be more economic since there is no need to buy a new TV set. However, it can result in a significant loss of quality: features such as enhanced image quality, higher image resolution, or advanced sound features might not be available.

10.2.3 Launch of Digital Television

A 2001 survey by the independent British Consumers' Association (see [CA 2001]) revealed that about one quarter of the population of the UK is not interested in switching from analog to digital television. Deterrents are the unwillingness to pay for DTV, unawareness of its advantages, and indifference towards more channels and new features.

Despite the results of this review, the British government sticks to its decision to switch off analog TV some time between 2006 and 2010. The policies of other European countries largely correspond to this (e.g., [Meyer and Fontaine 2000]): Germany has already passed an act about the discontinuation of analog television by 2010 ([MHP-Forum 2002]), and Austria is discussing a binding decree about the launch of DTV between 2004 and 2006 and the final switch off of conventional TV between 2008 and 2012 ([Standard 2003a]).

As far as New Zealand is concerned, hardly any specific information is available. A government discussion paper ([Hobbs and Swain 2001]) states, though, that DVB has formally been adopted as standard for DTV in 2001. Analog television might be discontinued between 2010 and 2015

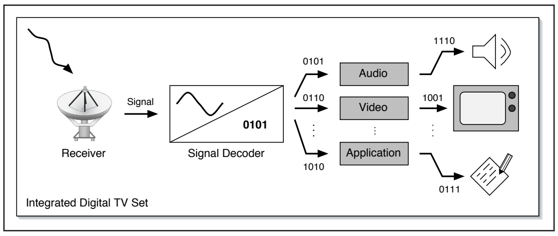


Figure 1: Architectural overview of an integrated digital TV set.

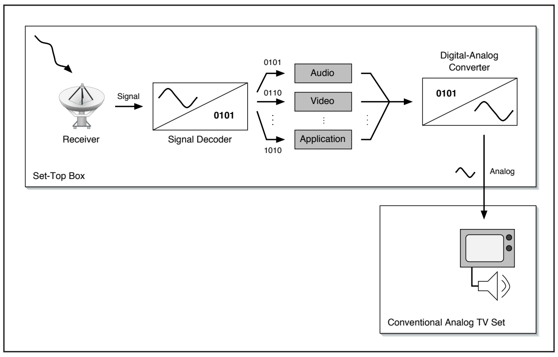


Figure 2: Overview of the architecture of a set-top box.

when the current licenses for UHF and VHF transmissions are due to expire. Concerns about the acceptance of DTV and other issues possibly delaying its introduction have also been raised in New Zealand.

In my opinion, DTV will *nevertheless* be launched gradually and users will *have to* cope with it – in spite of their dislikes for the new technology and their reluctance to the related expenses.

10.3 The Digital Video Broadcasting Standard (DVB)

DVB defines a digital video broadcasting system based on MPEG-2. The DVB Project was founded in 1993 to extend the MPEG-2 standard in order to form a complete and comprehensive system for future DTV environments. The DVB specification is published by the European Telecommunication Standards Institute (ETSI). However, the geographic

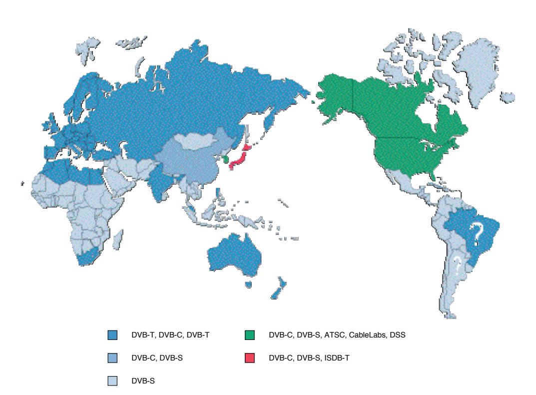


Figure 3: Geographic distribution of DVB. Map taken from [Peek 2002].

distribution of DVB is not limited to Europe but it has been selected by many countries worldwide as standard for their DTV networks. An overview is depicted in figure 3.

Although DVB uses MPEG technology, there is a different terminology: a transport stream is a *multiplex.* A TV program with all its elementary streams is called a *service* that consists of several *events* (the TV shows).

10.3.1 Transmission Channels

The DVB specification allows not only for numerous different carrier types for the multiplexes including traditional terrestrial, cable, and satellite transmissions, but also microwave frequencies and technologies such as ADSL, ATM (asynchronous transfer mode, a packetized multi-purpose network), Ethernet, and IP. In many implementations (e.g., satellite transmission) one multiplex has a bandwidth of approximately 40 Mbps. Depending on the video and audio quality, as well as the presence of additional data streams (see table 2), six to eight services can be fit into one multiplex. (See [Fairhurst 2001d,e; Morris 2002b].)

In order to enable interactive television, *interaction channels* have to be defined. Interaction channels are return channels from the consumer to the service provider. The return channel system consists of two components: the return interaction path from the consumer to the service provider, and the forward interaction path. In theory, the interaction channel can use a different transport medium than the broadcast channel, and even a different service provider could offer it.

An example: a consumer has DVB-S equipment to watch DTV, and the interaction channel is implemented with ADSL (probably by a different provider). When the user wishes to interact with the broadcast content, a request is sent over the return path of the ADSL network. The reply to the user's request is sent via the forward interaction path

Denomination	Meaning, Use
DVB-C	DVB Transmission via Cable (CaTV)
DVB-S	DVB Transmission via Satellite
DVB-T	DVB with a Terrestrial Carrier
DVB-SMATV	Single Master Antenna TV (professional)
DVB-MC/-MS	Multipoint Microwave Video
DVB-RCS	DVB with Return Channel via Satellite
DVB-RCT	DVB with a Terrestrial Return Channel
DVB-J	DVB Java Interfaces
DVB-MHP	The DVB Multimedia Home Platform

Table 1: DVB Synopsis.

which could either be the ADSL network or the broadcast medium – again depending on the implementation and on the type of service.

The implementation of the interaction channel can be straightforward: cable television, for example, could realize the return channel using a cable modem. For other media such as terrestrial or satellite transmission, that are pure broadcast media, it is more sophisticated. Despite technical difficulties return channels via satellite and terrestrial transmission have been developed (see [Paxal 2001; Fairhurst 2001g] and [Burow et al. 1998; Scalise 2001]). Alternative systems such as ADSL or modem connections can, of course, be employed as well.

Which technology is used for the interaction channel is completely up to the DVB network provider. Factors such as the required response time or the amount of data to be transferred are to be considered.

10.3.2 Data Transmission

Since one of the central aspects for the realization of user interaction and DTV applications is the transmission of digital data, the DVB specification describes five different ways to transport user data. These *profiles* are outlined in the following paragraphs. (See also [Fairhurst 2001e,f; DeLay 2001; O'Hagan 2001].)

DATA PIPING. In this model, the data to be delivered is spanned over several packets that are sent in the MPEG transport stream. A timing relationship among the data packets or a synchronization with other PES packets is not specified.

DATA STREAMING. In this approach the data is transmitted in a continuous stream, a PES. If there are no packets to send, it is common to insert empty packets. A stream can be synchronized (it is related to other PES packets), it can be synchronous (relational to a clock), or asynchronous (without any timing information). Supplementary information services such TV captions make use of data streaming (e.g., [Forbes 2001]).

Data streams have a benefit for service providers because the bit rate of the stream can be pre-defined, which makes the organization of the MPEG TS (transport stream) easier. A disadvantage is, though, that this might lead to a waste in bandwidth. If the full capacity of the data stream is not utilized, other streams or services could use it. However, from an organizational point of view this might not be possible because the bandwidth has already been assigned to the data stream.

MULTI-PROTOCOL ENCAPSULATION. Multi-Protocol Encapsulation (MPE) provides a LAN emulation which makes it possible to send data as on a packet switched network. MPE is based on the "Digital Storage Media – Command and Control" technology explained in section 10.3.3.

This method is recommended for internet data transmission (and similar computer network-based traffic).

DATA CAROUSEL. A data carousel can be seen as a circular buffer containing data, where only one element is accessible at a time. After having multiplexed the element with the transport stream, the buffer is rotated by one element (hence the designation "carousel") and the next element is sent, etc. This leads to a periodic transmission of the same set of data.

If some information is more likely to be requested than other data, the particular elements can be duplicated in the buffer. Consequently, they are transmitted more than once during one full revolution of the buffer, which results in improved access times for the user.

An example for a data carousel-based service is teletext. It is quite common that one has to wait for the teletext system to fetch a page. This is due to the architecture of the system: the requested page might not be the currently produced element from the cyclic buffer, and the system has to wait for the page to be transmitted again.

There are two ways to improve the performance of page retrieval. Since index pages in the teletext system are normally the most frequently requested pages, they can be inserted several times into the buffer. Therefore the time to retrieve an index page can be reduced. A further speed-up can be achieved by employing dedicated cache memory in the consumer device. The cache memory can store certain index pages or pre-fetch pages that were defined previously by the user.

Stream Type	Bit Rate
Video (usually between 4 and 6 Mbps)	5,000 kbps
Stereo Audio	250 kbps
Subtitles	50 kbps
Conditional Access	600 kbps
Service Information	300 kbps
Program Specific Information	546 kbps
Digital Teletext	754 kbps

Table 2: Typical stream data rates (see [Fairhurst 2001e]).

OBJECT CAROUSEL. The object carousel technology is very similar to the data carousel. It is optimized for (binary) data transmission, which can, for instance, be used to download application programs to the DTV receiver.

10.3.3 Digital Storage Media – Command and Control (DSM-CC)

The DSM-CC is specified in Part 6 of the MPEG-2 standard. It is a toolkit for defining control channels for MPEG-2 streams and has (at least partially) been adopted by the Digital Audio Video Council (DAVIC), DVB, ATSC, and other organizations. DSM-CC was designed to facilitate the implementation of features that are well known from VCRs: pause and resume, rewind, fast forward, etc. Moreover it enables data packet transfer within the MPEG-2 specification (e.g., [Bionic 2003]).

The Multi-Protocol Encapsulation introduced above is contained within the DSM-CC. Also MPEG-4's DMIF, the Delivery Multimedia Integration Framework, is an expansion of MPEG-2 DSM-CC ([Woodward 2000]).

The DSM-CC specification describes a client server architecture, where the consumer's terminal is the client and the service or content provider is the server entity.

One of the elementary concepts is a session ([Balabanian et al. 1996]). A session is a connection between two users that usually congregates the resources needed for this service. Resource allocation is dynamic, i.e. resources can be added or removed during a session. After all actions have been carried out, the session is terminated and all resources are released. Each resource is tagged with a session ID, which facilitates resolving the assignments of resources to sessions. It also permits the introduction of billing on the basis of the actual amount of transferred data or consumed services.

The data transport features in DSM-CC are designed for lightweight and fast performance. This can be explained by the fact that most devices using DSM-CC including set-top boxes currently have limited memory and storage capacities. Characteristics of the download protocol are the use of the sliding window method (as known from TCP), optional use of acknowledgements (ACKs) for delivered packets and flow control for broadcast media, and the mapping to the MPEG-2 transport stream for hardware multiplexing (see [DSM-CC 1997]). A server can multicast or broadcast to several clients simultaneously.

Except for data transmission, DSM-CC can also be used for the configuration of client devices. A set-top box, for instance, could utilize control and configuration messages to identify itself to the network when it is switched on. Furthermore information that is related to the structure of the network and the provided services could be retrieved. Thus, the client device could automatically configure itself. This user-network configuration is a separate part of the DSM-CC specification.

10.4 The Multimedia Home Platform

The Multimedia Home Platform (MHP) was established by the DVB initiative in 1997. It is a standard on top of DVB and was introduced to support the development of applications on DVB systems – independent of hardware manufacturers, broadcasters, or content authors. This means that a user terminal (digital TV set or STB) can receive and run applications, no matter what the vendor, service provider, or TV channel is (e.g., [Bergin 2000; MHP 2002a,b; Morris 2002c]).

Thus, MHP is a platform that enables applications to run in a standardized environment on top of DVB. It should be an interface between (interactive) applications and the terminal hardware on which they are running. This model is illustrated in figure 4.

Popular examples for MHP applications include electronic program guides (EPGs), information services such as stock or news tickers, games, and e-commerce systems.

10.4.1 Specifications and Requirements

MHP describes three different approaches, so-called *profiles.* The profiles target return channels with different bandwidths and the typical applications for these environments.

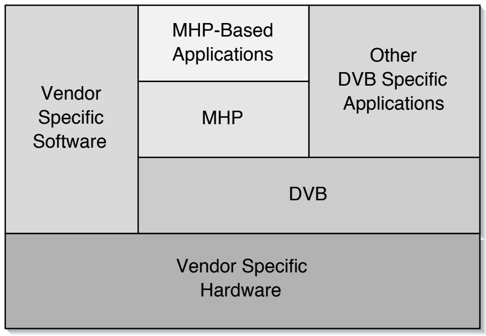


Figure 4: How the MHP interface is positioned within DVB-based systems.

The first profile only requires a simple telephone line as return channel, i.e. a modem connection. Since the network connection is very slow, downloading complete applications is unrealistic. Hence, the complexity lies within the STB (to keep network traffic low). This architecture represents merely enhanced television, true interactivity is not yet supported.

Profile 2 strongly resembles the first concept, but the aim is true interactive television. Not only data but even small applications can be transmitted over the network connection. In this profile there is a greater emphasis on the software aspects and on the question how interactive applications can be embedded in the system.

The third profile focuses on broadband network connections such as cable modems and faster technologies. These enable a whole new range of data intensive applications and provide support for internet-based actions and content delivery.

The three profiles described above finally yielded two specifications. MHP 1.0 comprises the first two profiles that use basically the same technologies. Its designation is enhanced- and interactive-television. The third profile is contained within MHP 1.1 with a focus on enhanced and interactive television as well as internet access. The two versions of MHP are very detailed standards with a size exceeding 1,400 pages.

The different profiles require, of course, different hardware configurations. [Smith-Chaigneau 2001] mentions a 300+ MHz CPU and 128 MB RAM as an industry recommendation for MHP Profile 3.

It is beyond the scope of this thesis, but it should be noted that the development of hardware for digital television devices might soon see trends similar to the ones in PC hardware business. As more powerful hardware becomes available and gets cheaper, vendors use it in their STBs and integrated digital TV sets. At the same time applications progressively require improved computing power, more memory and storage space. Thus, in future not only the size of the screen will be important but also other features such as the CPU generation, the memory size, or the capacity of the hard disk.

In a worst-case scenario users might be forced to upgrade their TV hardware regularly (or even buy new equipment) in order to be able to access the latest DTV applications. This circumstance opens a whole new market that seems to be quite similar to the PC business.

10.4.2 Architecture and Design

The architecture of MHP can be seen as a model consisting of several layers. The three most important layers are: system resources, the system software, and applications.

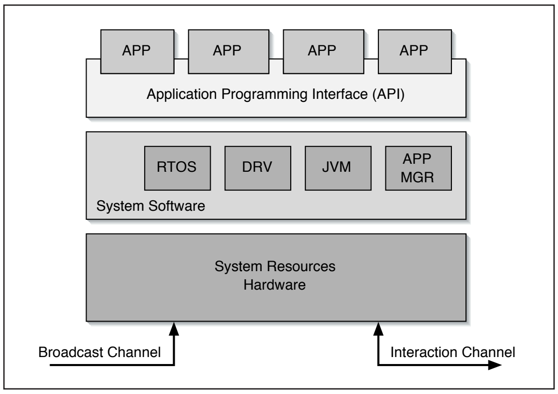


Figure 5: Overview of the MHP system architecture.

Resources are hardware components such as the MPEG processor, the CPU, memory, or the graphics system. As in common computer systems, the layer with the system software manages the hardware resources and provides an abstract model to access them. The system level contains elements such as a real time operation system (RTOS), the Java Virtual Machine (JVM), device drivers, and an application manager that handles the complete life cycle of MHP applications (see [Wipro 2001]).

The specification of the hardware- and system software-layers is generic. Therefore more or less vendor-specific implementations of these two layers are possible.

The third layer consists of Java application programming interfaces (APIs) and application programs running in this environment. The APIs are strictly standardized and enable the development of applications that are independent of the underlying two layers. A simplified model of the architecture is depicted in figure 5.

Applications written for MHP compliant devices are programmed in Java (e.g., [Wipro 2001]). Using Java technology as programming language and application programming interface (API) is a big advantage. All that is needed is a PC and the Java SDK, which makes it relatively inexpensive to start application development. However, the problem is to test the complete application in a real life environment because currently hardware conforming to MHP is not yet availble.

Writing a Java application for digital television (DTV) or rather MHP can be quite different from writing conventional, local Java applications or web applets. Constraints such as the limited availability of bandwidth and memory, optimizations for short loading times, interdependencies of classes and external modules, and the small footprint of client devices have to be considered for DTV applications.

MHP applications are transmitted in a DVB transport stream or via an IP-based connection. Therefore the use of MHP not restrained to DTV, but it can also be employed in intranets and on the internet. Thus, potential MHP client devices are not only set-top boxes but also networked computer workstations and ubiquitous computing devices.

10.4.3 Application Programming Interfaces (APIs)

MHP's architecture relies on five Java APIs: Sun's Java Media Framework (JMF); the Java TV APIs; DAVIC, a framework defined by the Digital Audio-Video Council (e.g., [Thompson 1999]); HAVi-UI, a user interface API for home audio video interoperability; and the DVB-MHP API.

JMF. The JMF labelled javax.media is an extension to the Java Core that makes the use of audio, video, streaming and other media in Java applications and applets possible ([JMF 2002a,b]). JMF also supports internet protocols such as RTP and RTSP as well as the Quicktime Streaming Server software.

An interesting development in this area is IBM's MPEG-4 decoder for the JMF; it was written completely in Java.

JAVA TV. The Java TV API, javax.tv, is a project undertaken by Sun together with companies and organizations from the DTV industry (see [Java-TV 2002a,b]). The framework provides access to all major features of a DTV receiver including audio and video streaming, data channels, control over the tuner and the on-screen graphics. It also offers functions for media synchronization and for the management of the application life cycle.

Java TV relies on the interface provided by the Java Media Framework described above.

DAVIC. The API (org.davic) defined by the Digital Audio-Video Council comprises a set of tools for digital television environments. It includes access to the DVB service information and to the conditional access system. Moreover, private and user data sections from a continuous MPEG-2 stream can be filtered (see [Piesing and Löytänä 1999]).

HAVI-UI. The HAVi-UI framework, org.havi.ui, was developed by the Home Audio/Video Interoperability (HAVi) Consortium ([HAVi 2002]). It is designed to be used for user interfaces of consumer electronic devices.

HAVi-UI is an extension of the Java AWT package that is capable of determining the interface facilities of the client device. Furthermore it makes it possible to read the user's input or display objects on the user's screen.

DVB-MHP. The top-level API for this collection of frameworks is DVB-MHP as specified in org.dvb. The MHP API is responsible for return channel connection management, access to persistent storage, security mechanisms, event handling, application launching and signalling, and suchlike operations.

10.4.4 MHP Market Launch

The introduction of the MHP Standard is planned in many countries and, in fact, it has already been launched in some European countries: Finland, for example, has used it in its DTV initiative since August 2001, and in Munich, Germany, a pilot project of MHP over DVB-T is being carried out.

Amongst others, Australia, Korea, and Singapore have selected MHP as API of choice for their DTV programmes, while China, for instance, is seriously considering its commitment (see figure 6). At the moment, New Zealand uses the proprietary OpenTV system, but once the implementation of the DVB network has begun, most probably MHP will be employed.

MHP has strong support from TV companies and all related industries. Members of the MHP Implementors Group include:

- TV channels and network providers such as ARD, Canal+, Nine Networks Australia, NTL, ORF, RAI, RTL, and the Singapore Broadcasting Authority;
- hardware vendors such as Loewe, Panasonic, Philips, and Sony;
- content providers including the Bertelsmann Group;
- other companies such as Nokia, SES/Astra, Sun Microsystems, and VW;
- several government organizations and universities.

There are some other, independent groups such as the MHP Experts Group, the MHP Action Group, or the German MHP MarCom. They try to get companies that are interested in the standard started and support them.

10.5 A Different Approach to Digital Video

The sections above describe the traditional approach to digital video broadcasting that includes DTV and interactive television. Now a different approach will be concisely presented.

KiSS Technology, a Danish manufacturer of DVD players recently introduced a new appliance, the DP500. Basically, it is a DVD player with several advanced features. However, it also implements technologies that do more than just play back DVDs. The device supports MPEG-4, DivX 4 and 5, and allows for 10/100 MBit Ethernet connectivity (e.g., [Standard 2002; KiSS 2002; Lustrup 2002]).

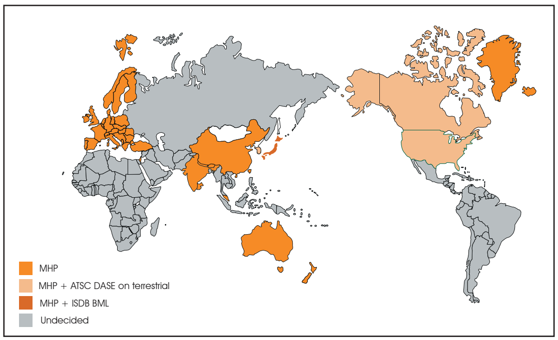


Figure 6: Geographic distribution of MHP. Source: The DVB Project Office ([DVB 2002]).

The DP500 consists of specially designed hardware (see [Sigma 2002; Sigma 2003]) and general purpose software. The interesting aspect is that the software platform employed is Embedded Linux, a version of the Linux operating system particularly for integrated systems and consumer electronics ([ELC 2003]). This makes it rather easy to add new software components and hence new features to the system.

As such it is a "networked media player" ([Lustrup 2002]). The device can be connected to a high speed network, and technologies such as video on demand (VoD) or also broadcast DTV services via Ethernet may be used. However, the implementation is quite different to a DVB system, for instance. DVB specifies details about the network, the transmission media and the transmission process, whereas this network media player relies on proprietary implementations based on the underlying protocol – IP.

Another approach to make the DP500 ready for DTV is changing its software platform. The DVB standard permits Ethernet networks to be used as transport media. Therefore the DP500 could be applied as a DVB conformable set-top box, if a corresponding software stack were installed. In case a Java Virtual Machine can be installed, even MHP might be utilized. Of course, major modifications in the underpinning APIs would be necessary.

A problem is, though, that there is no freely available specification of the product and interfaces are not openly defined. Thus it is hard to say if modifications are possible at all.

However, this specific product should only serve as an example that there are already devices available that are based on the latest technologies such as MPEG-4. Furthermore the network interface makes it possible to introduce true interactivity, VoD, and other innovative features.

10.6 Conclusion

This chapter provided a brief introduction to digital video broadcasting and the underlying technologies and standards. MPEG-2 is the de-facto standard for video compression and coding in the DTV sector. DVB is already used in many countries around the world and it is rapidly becoming the most important specification for digital video broadcasts. Additionally, the DVB initiative attempts to establish the Multimedia Home Platform as an independent standard for application development on DTV devices. However, it is a relatively new standard, and the market acceptance is not clear, yet.

Apart from the enormous standardization efforts made by groups such as the DVB initiative, several other proprietary systems emerge. An example is the DP500 DVD player with an integrated network interface that makes it potentially possible to handle video streams from intranets and the internet. This is certainly an interesting alternative. Such a network would be cheap and relatively easy to implement provided a high speed network connection is available.

Vivid – A Design for Active Digital Video Broadcasting

11

11.1 Introduction

This chapter introduces a concept for active digital video broadcasting. First, the basic idea is described together with some characteristics of the proposed system, and the relation to active documents is outlined. The subsequent section, a central part of this chapter, presents a number of usage scenarios and application areas for the technology. Furthermore the role of the different kinds of information in the system is detailed, and finally a few economic aspects are explained.

11.2 The Basic Idea

Do questions such as *"In which movie have I seen this actress before?"* or *"What is the title of the song in the background?"* sound familiar to you? Today's digital video systems including DVDs, digital television (DTV), and computer-based video players are not capable of answering these questions.

This chapter proposes Vivid – a new system for **V**irtual **I**nteractivity in **VID**eo broadcasting environments. Vivid extends the conventional broadcasting methodology in a way that allows users to find answers for their questions while they are actually watching a program on television. Moreover, Vivid lets consumers request detailed information on a scene they are watching at the moment. Among other information, this set of data also includes names of actors and sound tracks.

Additionally, the proposed system can provide functions that make it possible to extract certain segments or objects from the content. This feature is particularly useful for downloading a short video clip or obtaining the audio track of a broadcast.

This project employs recent techniques for content description and designs a new system for digital video broadcasting environments that makes it possible to provide explanations for many of the user's questions. In doing so, it offers enhanced interactivity to the consumer.

Vivid delivers knowledge in an uncomplicated way, quickly, and when the user requires it. In this case, knowledge management (KM) is employed in a rather untypical domain: although KM has successfully been applied in computer-based environments, digital television and similar video broadcasting technologies are quite new application areas.

11.2.1 Characteristics of the System

The proposed system describes a synchronized application. This means that the transmitted additional information is related to and relevant for the content that is displayed at the moment. However, also some asynchronous elements such as (file) downloads or requests for further information can be included.

From the system's point of view, the content is made up of objects. These objects can be elements such video segments, audio clips, and still images. For every object a rich set of metadata is retained and transmitted to the user. This makes it possible to answer the users' questions. Additionally, the transmitted metadata enables a whole range of other, new applications.

The basic idea of the system is not strictly limited to digital television and traditional video broadcasting technologies but can also be used with DVDs, for instance. Moreover, the idea of the system is not confined to a television set as consumer device, but also computers and a whole range of mobile devices are understood as potential client devices. (See also figure 1 in chapter 12.)

11.2.2 Relation to Active Documents

Vivid makes use of a variant of active documents as they are described in [Heinrich and Maurer 2000] (see chapter

6). In the original model, answers to users' questions are either given "online" by the system or "offline" by experts. The concept works best when a large number of users access a limited amount of documents. In this case, after some time the same questions are asked again and again, and the number of new questions decreases rapidly.

However, the same assumption cannot be made for digital television systems. This can simply be explained by the fact that the amount of content broadcast every day is far too big in relation to the user-base. Therefore too many new questions might be asked, and eventually the system might become unmanageable. Hence, the proposed system chooses a different approach.

In a first step, those questions that are most likely to be asked have to be identified. This can be done in an empirical study, for instance. Depending on the content and the context, the questions will most probably be quite similar and rather general. Examples for typical user requests are *"Who is this actress?", "When did that happen?",* or *"Give me more information on this event."*

Then, answers to these questions have to be generated for all scenes of a TV show. This process can be manual or include partially or fully automated techniques. Moreover, further references and links to external resources or related information are added.

The information collected in the second step is transmitted together with the content to the client device. When users have a request, Vivid presents this pre-compiled set of data. Probably in most cases the majority of questions can be answered with this information. If the consumer wishes to get in-depth information, more details can be obtained from references to external resources that are offered by the proposed system.

Clearly, this technique is not as flexible as the original active document concept. However, the pro-active dissemination of a well-considered, though limited, set of information is an attempt to adjust traditional active documents to the conditions imposed by the DTV environment. As the examples below show, this approach can be quite satisfactory.

11.3 Usage Scenarios

Although the basic intent was to create an environment that makes it relatively easy for the user to find out more information about *movies,* the paradigm can be extended successfully to other domains.

Several prospective usage scenarios including movies, news and documentaries, sports, children's programs, and music television are addressed in the following sections.

11.3.1 Movies

A person is watching a movie on TV and wonders where she has seen this particular actress before. She uses a pointer device built into the remote control and selects the actress on the screen. A window pops up and presents information on the actress: her name and role in the movie, her real name, other data, and a reference to a specialized movie database such as the Internet Movie Database, IMDb ([IMDb 2002]), as well as link to her web-site. If the viewer wishes to know more, she simply selects a function to query the specialized database or to go to the actress's personal website.

In a similar way the viewer can find out which other movies the director has directed, who has written the book or script, whether this film has been awarded, etc.

Another example focuses on the sound track of the movie: the user likes a particular song and would like to have it. She requests information on this scene, a window on the screen pops up and lists all noteworthy elements of the scene. It includes all actors and audio clips. Now the user knows the name of the song and the artist and can even choose to download it instantly to the set-top box, computer, etc.

The example in figure 2 shows a scene from the movie *"2001: A Space Odyssey."* Vivid superimposes the (blue) window on the conventional TV broadcast and uses this area to present the content-related information. It tells the user the names of the two actors that occur, and the title of the sound track is given. As a commercial application, a reference to an online book shop is provided.

Now, the user has the possibility to download the sound track of the movie or buy the book to the film online. These actions are enabled by the proposed system.

11.3.2 News and Documentaries

News and documentaries can sometimes be quite difficult to follow if one is not familiar with the context, domain specific facts, or the historical background. In this case references to related "articles" or background information might be very useful. Basically, this is an analogy to many news services found on the internet.

In a news broadcast the proposed system can be used to offer more specific details for interested viewers. A news item about the Columbia space shuttle disaster, for instance, can be supplemented with a graph showing the flight path, a chronology of the accident, related news stories, etc. A hyperlink to the Nasa website complements the report. An example of a newscast is depicted in figure 3.

Documentaries can use similar features: imagine a school class watching a film about World War II. The teacher wants to explain some events in detail and therefore he requests additional information. VIVID could offer a variety of audio documents, other video clips, maps, text documents, etc. The teacher selects a historical map, and has it displayed on the screen. Alternatively he could download it and, for instance, print it out.

11.3.3 Sports

Some features of the active digital video broadcasting concept can also be applied for sports programs. VIVID can, for instance, make related reports and additional information available.

A rugby match, for example, can be rounded off with results of previous games, information about games that are going on at the same time, etc. In case of a championship, the teams' results of previous years can be looked up as well.

Figure 1: Some Microsoft Agent characters. Image sources: [MSAR 2003a-c].

The broadcast of a skiing race can be enhanced with training results, a downloadable map of the ski-run, and an interview with the athlete. To emphasize economic aspects, additional functions can include merchandising and booking tickets for future events. See figure 4 for an illustration.

11.3.4 Children's TV

The examples above are largely text-based. In children's TV programs, a different approach has to be found because the assumption that all viewers are able to read might not be true. The need for an alternative concept in this case is quite challenging.

For the proposed system, a technology similar to Microsoft's Agents is suggested. An agent is a user-interface element that interacts with users. Instead of confronting users with pop-up windows, the agent asks questions acoustically or in balloons, and so there is a way to communicate with a more human counterpart (e.g., [Bell 2003; Zimmermann 1997]). Three popular Microsoft Agent characters are illustrated in figure 1.

The agent could speak to the children and tell which actors appear in the scene: "The little boy is Johnny Smith. The animal you can see is an elephant." Alternatively, the agent could ask the consumer to click on an object, and then it reads only information about this very object to the user.

In this approach children do not have to be able to read because an agent will talk to them and present the available options. In addition to this there can also be a number of easily recognizable symbols such as a note for music, a brush for painting, or a blackboard for school.

The descriptions offered by the proposed system should be easy to understand and can be made in a playful way. Examples include graphical depictions of complex relationships, acoustic explanations of background information, or spoken descriptions of characters.

11.3.5 Music Television

In music television some of the proposed functionality can be useful as well. Especially for this field, downloading content can be quite attractive.

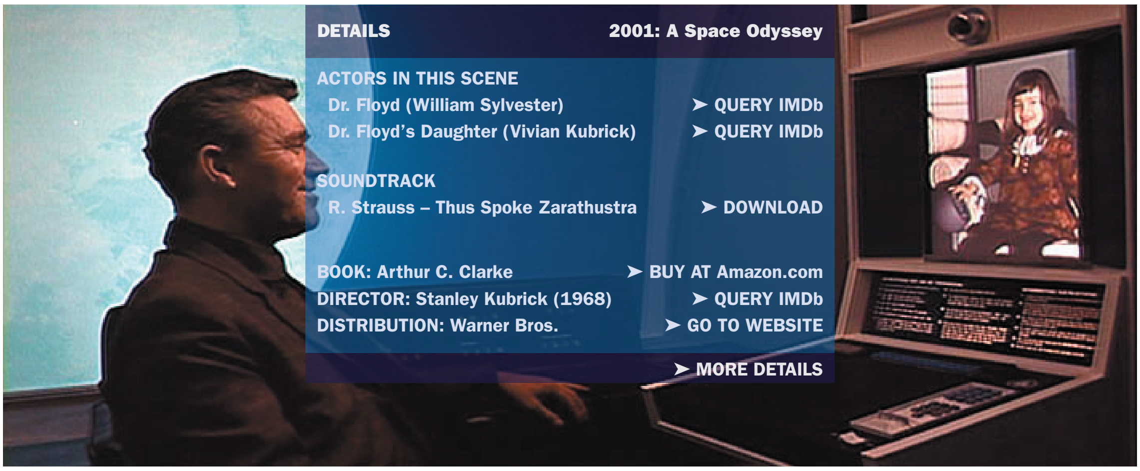
When consumers are watching a music video, they can choose to get detailed information on the video or the artist, obtain the band's discography, etc. Furthermore, there might be a possibility to store the complete music video on a local storage device or download the song only (without the video).

Some aspects of this scenario are detailed in section 11.5 below.

11.4 Information in the System

The system stores and processes three different types of information: content, metadata, and digital rights. While the meaning of content is quite clear – videos, sound, previews and trailers, etc. – the sense of digital rights and especially metadata needs further explanation.

An example for the different kinds of information is shown in figure 5: every object that occurs in the scene has a set of data attached that, theoretically, includes an object specification, metadata, as well as digital rights and intellectual property information.



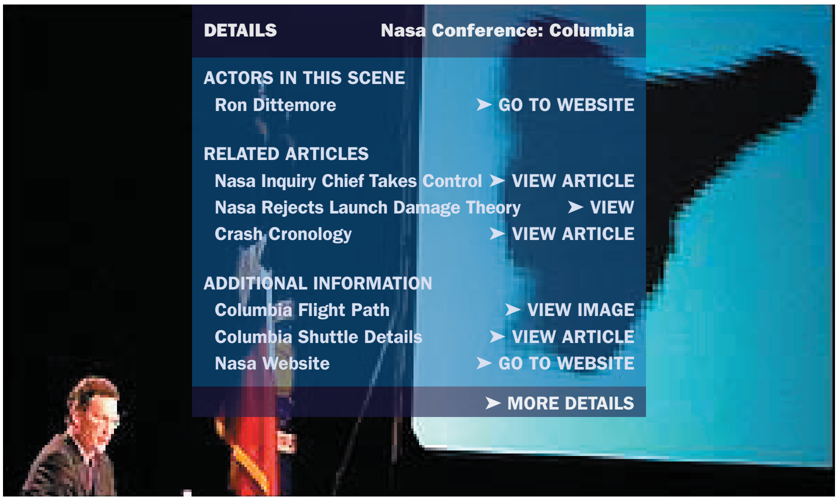


Figure 2 (top): A scene from *2001: A Space Odyssey.*

The user requested information on the scene, and VIVID superimposes a blue box that contains details. There are two actors and a song playing in the background. Additional information includes, for instance, the author's and director's names. External resources such as IMDb or Amazon.com are referenced. Original image source: [DeMet 2001].

Figure 3 (left): A scene from a news broadcast on the disintegration of the Columbia space shuttle. Original image source: [BBCi 2003].

If the user is not familiar with the events, he can choose to view a crash chronology (additional information) as well as previous articles and TV programs (related articles).

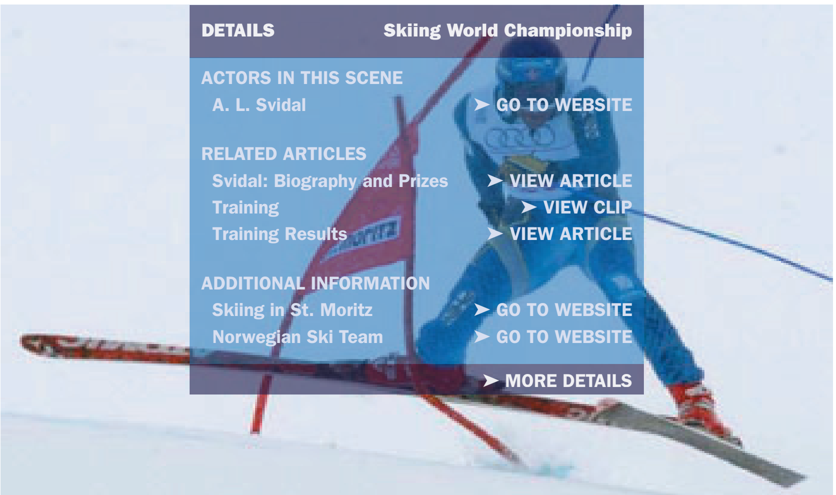


Figure 4 (bottom): A scene from the skiing world championship 2003 in St. Moritz. Original image source: [Standard 2003b].

In this example the user can obtain personal information about the sportsman, watch the training or simply view the training results, and get additional information.

The users can also go to the website of St. Moritz and possibly book their skiing holidays online.

The large amount of metadata that is required for the efficient operation of the system can be generated with fully or at least partly automated techniques (see section 11.4.4).

11.4.1 Content

Content is usually audio or video material, but it can be any kind of multimedia data. Content can be an item such as a complete movie or only a certain scene of a it, it can be a separate audio track, a film preview or a trailer, an animated graphic or a still image.

The design of the proposed system is open, it does not impose any restrictions on the type of content and its encoding. This means that encoding standards such as MPEG-2 and MPEG-4, but also proprietary specifications including D1, D5 and Sony's Digital Betacam can be used (e.g., [Currier 1995; Utz 2000]).

Which storage and retrieval methods are used is also left undetermined in order to facilitate the integration of Vivid into existing architectures. Some storage technologies include keeping the content in a database, saving it in a filesystem structure, and retaining it in external resources such as tape archives.

Content does not necessarily have to be stored within the organization. It is also possible to retain references to external sources that provide the actual content. See also section 11.5.1.

11.4.2 Metadata

The additional information that is offered by the system including names of actors and information about background music is retained as metadata to the content. Also references to related material and additional resources as shown in the usage scenarios above are stored as metadata to the corresponding content.

Metadata can be collected in different levels of granularity (see section 4.3.2 about multilevel descriptions of metadata). The most elementary information is the name of the TV program, its start time and the duration. A short summary as found in TV guides and teletext is usually also available. This set of data can roughly be seen as bibliographic information.

Another kind of metadata are structural connections between objects. If, for example, two scenes are set in the same location, there is obviously a certain relationship between the two scenes. If two actors appear in the same scene at the same time, a relation between them probably exists. Metadata on a structural level can be used to analyze content, to describe its structure, or even to generate new knowledge.

On the lowest level of content description, characteristics such as color histograms, Tamura textures ([Tamura et al. 1978]), motion vectors, shape descriptors, or features describing sound can be collected. This data can be employed to enable content-based retrieval and querying based on the similarity of features (e.g., [Atnafu et al. 2002], see also section 12.3.2).

Although previews and trailers can also be seen as metadata that is attached to a certain piece of content, this method is rather disadvantageous. A preview might have a separate set of metadata associated with it, and entirely different rights and permissions might apply. Therefore an approach seems favorable where previews are independent objects that are only related to the original content.

Conceptually, the description of Vivid architecture of the system does not imply a particular metadata standard to be used. MPEG-7, Dublin Core, and other technologies can be employed. This approach has been chosen in order to enable reuse and integration of existing metadata-bases. Furthermore a system can be chosen that suits the needs of the particular service or system environment best.

11.4.3 Digital Rights

The digital rights section, in fact, comprises two separate parts: digital rights and intellectual property information. An intellectual property record basically says who owns content, who distributed the content, and probably contains a license. Digital rights data, on the other hand, specifies which operations can be carried out with the content and which ones are prohibited.

The system makes use of both digital rights and intellectual property information, the latter being more important. It is used to determine whether a certain object may be extracted from the content stream, if it can be downloaded to a local storage device, or if it may be duplicated.

11.4.4 Generating the Information

The content as such is usually obtained from external sources such as film distributors or content authors themselves. To a certain degree, content is also produced within the organization. This is especially true of news programs.

Metadata and digital rights are merely *consumed* by the proposed system – Vivid does usually not *produce* them. Metadata and digital rights originate from external providers such as film distributors, but they can also stem from internal sources. Optionally, they can also be generated by a dedicated component within the organization.

Metadata from internal sources stands for entries in teletext systems, electronic program guides (EPGs), closed captions or other internal databases. This kind of information is very often available in television environments or video archives.

For the (optional) generation of metadata, technologies similar to the IMKA project can be utilized (see section 4.4.1 and [Benitez et al. 2001a] or [Zhong and Chang 1998]). The IMKA project uses object separation and MPEG-7 descriptors to specify content-related characteristics of media. IMKA attempts to provide content descriptions on semantic and perceptual levels with the aim of enabling content-based retrieval.

The process of generating new metadata as part of a proposed system architecture is also briefly covered in sections 12.3.2 and 12.2.5.

11.4.5 Presenting the Information

In the usage scenarios explained above, information is largely text-based. This sub-section shows some examples of how the proposed system can offer information to the user in alternative ways.

A rather traditional way of representing the information is to display all scenes of a program in a chronological order. Every scene is attached to a timeline, and the user can simply get a brief overview of the content. The model can be made a bit more complex by introducing a multilevel timeline: one level is used for video content, one level is used for sound tracks, one level is used for speech, etc.

The complete movie can also be presented in a hierarchical view, viz., as a tree. The root of the tree is the movie as such. It contains of a number of nodes, the scenes of the movie. Each node contains several other nodes: one for video content, one for audio content, etc. Leaves are attached to the sub-nodes. The leaves are the concrete pieces of content – a short video clip, the audio track of the scene, etc.

A more sophisticated approach is to represent the information in a semantic map. The structural information that can be collected as metadata is used to generate a map (see section 11.4.2 above). If two objects appear in the same scene at the same time, or more generally, if there is a spatial, a temporal, or a spatio-temporal relation between two objects, they are connected in the map with an edge. If a connection occurs several times, the a higher weight is assigned to the edge. Hence, the map can be used to find semantic relationships between objects and to highlight the most important connections.

11.5 Economic Aspects

Both the implementation and the ongoing organization of VIVID cause costs. The metadata archive, for instance, has to be maintained, new data might have to be acquired, etc. Therefore there must be some economic aspects that jus-

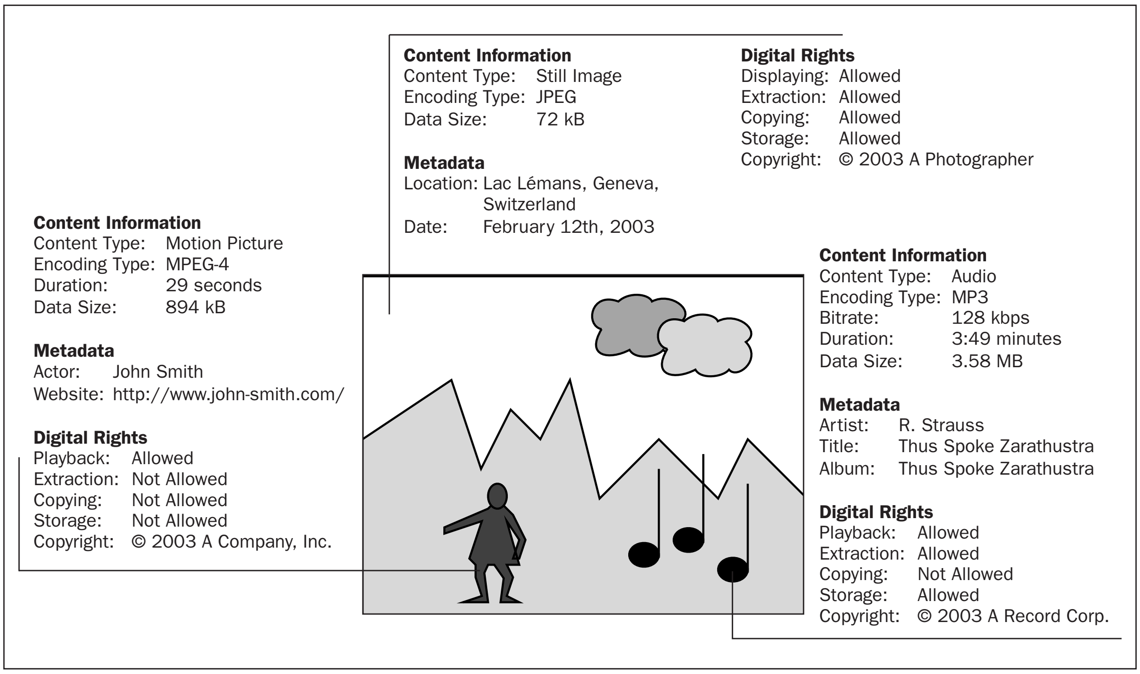


Figure 5: Example for metadata and digital rights in a scene with a moving object, a still image, and a sound track.

tify the concept; there must be an incentive for an organization to implement such a system.

The example with the background music introduced above (see section 11.3.1) can be used to explain a feasible, simple business model for Vivid. In this example, the viewer hears a song in a movie on TV and would like to get it. Traditionally, the first step for a consumer would be to find out what the name of the song and who the artist is, and on which album to find it. This can sometimes be quite demanding. Then, the user tries to find a "web shop" that has the album in stock, or offers it for download, and probably ends up buying a complete album although he only wanted to have one song ...

This is far too complicated, and most people are either deterred by the effort or have already forgotten the song by the time they go to a record store. With the proposed system, though, the user can buy the song immediately and download it without any inconvenience. On the one hand, this is good for the content provider because without the new technology the user might probably never have bought the song. On the other hand, it is good for the users because now they can easily access the content they really want to have. – From this perspective, the additional time that has to be invested in the creation of the content and the metadata is justified.

Thus, a possible business model is based on a cooperation of content distributors and TV channels. The TV channels offers to download content in its broadcasts: songs, videos, certain sequences of a movie, etc. can be obtained via the TV channel. Besides, references to related online shops can be integrated, for example, a hyperlink to Amazon.com to buy the book to a film. The TV service provider gets a sales commission for each piece of content that is sold via the television network.

11.5.1 An Implementation of the Model

The example above also shows one of the major difficulties: the organization implementing Vivid has also to allow for an infrastructure to store and maintain sound tracks and similar data. For many broadcasting service providers this might be too complicated or too expensive.

Thus, a particular implementation can make it easier for the TV channel: downloadable content does not necessarily have to be stored in an internal archive, but it can be retrieved from the original content distributor at the time when it is requested. This means that the television service provider does not have to take the organization, storage, and maintenance of this additional content into consideration.

The described business model and implementation bring advantages for both the TV channel and the content distributor or content author. Content distributors suddenly have another distribution channel, and there are new consumers that might otherwise not have bought the specific book, CD, video, etc. The TV channel, on the other hand, can offer an additional, new service at relatively little extra effort, and has the opportunity to cover the cost that the new system causes.

11.6 Conclusion

Vivid introduces a concept to make conventional video broadcasting more active, to encourage the user to interact with the content, and to facilitate the access to knowledge that might otherwise be difficult to obtain. The proposed idea can be applied to virtually any kind of video content – from movies to documentaries and newscasts to children's television. Consumers can use the advanced features not only for obtaining information on scenes or TV programs, but they have also the ability to download content and can reuse it. This makes the application a useful addition to conventional television services.

The basic requirements for the system are quite open in order to enable a certain flexibility. Any kind of content encoding technology, metadata standard, or digital rights specification can be employed. Also the output channels and potential consumer devices are not strictly pre-determined. Networked computers and mobile devices can be used along with digital television sets or set-top boxes. This diversity is also reflected in the definition of the generic system defined in the next chapter.

12 System Architecture

12.1 Introduction

This chapter presents the architecture that is used to implement Vivid, the system described in the previous chapter. The first section demonstrates the architecture of a generic system. It is a rather abstract model that contains all necessary components but does not specify details and their characteristics.

In the subsequent sections two concrete systems based on the generic description are defined. The first architecture uses MPEG-4 with computer networks and mobile devices as target group. The second approach is based on MPEG-2 and is destined for digital television.

12.2 The Generic System

The proposed system can be understood as an add-on to existing environments. This section explains how the suggested design can be incorporated into an existing TV broadcasting process and which connections between the new elements and the existing ones are established. A concise overview of the generic system and its main elements is shown in figure 1.

Further sub-sections describe the individual components in greater detail. First, design goals are outlined. Then, aspects of storing the different kind of information are described. Based on the system architecture and communication model specified in the next section, the system's major components are described.

12.2.1 Design Goals

The system is designed to reduce the cost of maintenance, to permit the use of existing resources, and to make future modifications relatively uncomplicated. The key design goals are presented in the next few paragraphs.

MODULARITY. Each part of the system is an independent module. This makes it possible to exchange one component with a similar, yet different module. Instead of an MPEG-7 metadata-base, for instance, a Dublin Core-based setup can be used.

INTEROPERABILITY AND REUSE. The system should allow interoperability with other systems and facilitate the reuse of existing components. It should, for example, be easy to reuse a metadata-base that is already available.

EXTENSIBILITY. The architecture of the system takes future extensions into consideration. New coding schemes or transport channels, for example, can be added to the system as new components without the need to change the complete architecture. Also new databases and storage facilities can be added relatively easily.

FLEXIBILITY AND ADAPTABILITY. Although a concrete idea about a certain set of features of the system is presented (see chapters 11 and 13), many of them are optional. Environments with limited bandwidths, for instance, can include only a basic set of functions. Thus, the approach enables the adaptation to different underlying technologies in the broadcasting process.

From a technical perspective, limitations to an implementation are predominantly made by the assigned bandwidth and the availability of a return channel. Therefore terrestrial DTV, cable television with interaction channels, and IP-based environments will result in different implementations with different characteristics. They may range from passive information dissemination to highly interactive applications.

The system should provide the flexibility to make these diverse implementations possible.

DISTRIBUTABILITY. The design should include arrangements to enable a distributed system. Therefore the internal structure of the system is basically the architecture of a distributed system. This permits the physical and even geographical distribution of the systems, which is rather common in television broadcasting environments.

ACCESSIBILITY. The proposed system architecture should permit different access rights for each module of the system. This means that the content archive, for example, may only be accessed by core system modules, whereas the metadata-base can even be queried by external systems that generate electronic program guides or perform similar operations.

12.2.2 Data in the System

This section briefly discusses several methods for storing data in the system. Moreover, aspects of organizing metadata and digital rights are outlined.

CONTENT. Content is often stored in large tape/media archives, in files in a structured filesystem hierarchy, or in special databases. The storage architecture is usually predetermined, and other systems have to be adapted to this conception.

METADATA. There are various ways of retaining metadata. It can, for example, be stored together with the content. An example are audio formats such as MP3 where metadata can be stored in the header of the file. Another approach is to save metadata in separate files. For every content item exactly one metadata file exists. If there is no metadata, the file is empty.

The third and preferable approach is to use a database as the metadata repository. Advantages include the availability of functions such as full text search or advanced queries.

In order to be able to relate metadata in the database to a certain piece of content, it has to be indexed. Two different techniques for indexing detailed metadata sets can be identified. Metadata can be directly attached to content objects so that every metadata record is linked to a specific object. In this case, the index is related to an object identifier.

The alternative approach is to connect metadata with content on the basis of timing information. Every metadata record is supplied with a timestamp and can be associated with the content via this timestamp.

An advantage of the timestamp-based system is the fact that metadata can be generated fully automatedly from storyboards and screenplays. A disadvantage is the dependency on the time codes. If the timing information of the content is modified because one scene is removed, for example, the metadata will become inconsistent.

The second approach is suitable for content encoding mechanisms that do not support independent objects (MPEG-2 for example). Although this method is quite different from the first approach, the user will most probably not be able to perceive the difference in a consistent implementation.

DIGITAL RIGHTS. It is customary to understand digital rights as a special kind of metadata. From this perspective, the same two concepts for managing digital rights can be exploited: an object-based and a timestamp-based approach, which results in the same benefits and weaknesses.

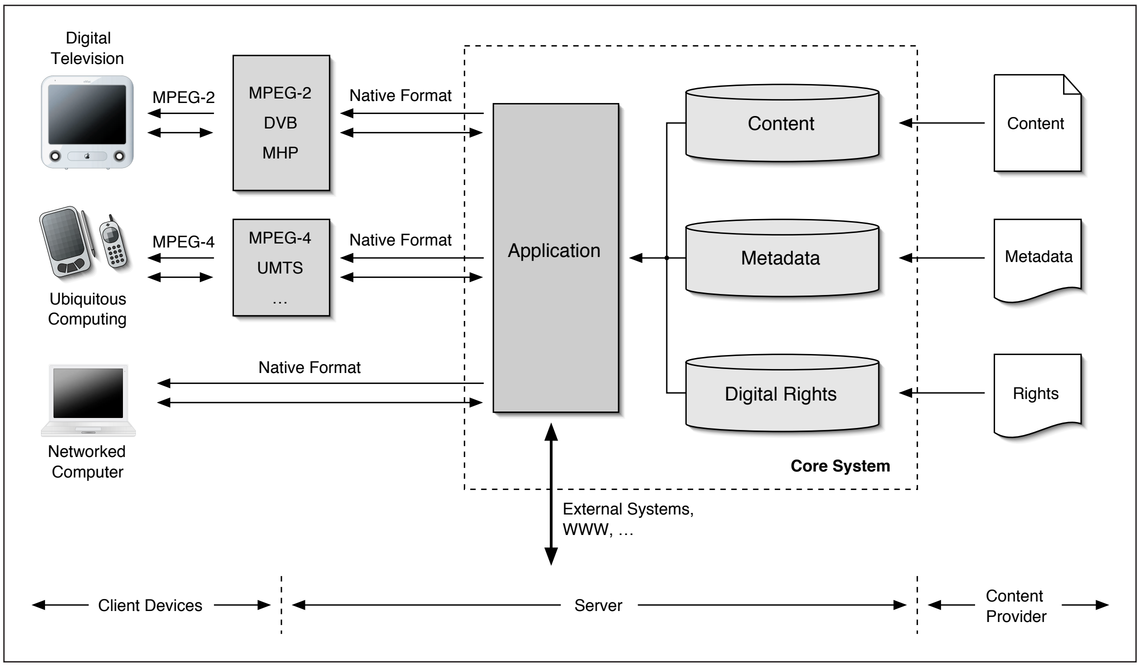


Figure 1: The system overview of the generic architecture with the core system, input and output components.

12.2.3 System Architecture

The implementation of Vivid follows a classic client-server paradigm. As implied in figure 1, the broadcasting service provider can be seen as the "server", and the consumer devices such as digital TV sets or computers as clients.

The server consists of the core system, an input and an output component. The core system (depicted as a dashed rectangle in figure 1) contains repositories for metadata and digital rights as well as the Vivid application modules. It can also include an archive for the storage of content (cf. section 11.5.1). The system's input component converts, filters, and organizes the incoming data and stores it in the archives of the core system. After the components of the system kernel assemble the information that is relevant for a video broadcast, the output module transmits it to the client devices (outlined on the left-hand side of figure 1).

The clients shown on the far left-hand side of figure 1 are connected to the server via networks of different topology and architecture. They comprise as diverse technologies as cable and satellite networks for DTV broadcasting, wireless LANs and UMTS (Universal Mobile Telecommunications System, [UMTS-Forum 2003]) for ubiquitous computing, and Ethernet or ATM for networked computer workstations.

The communication between the server and its clients is based on two logical transfer models: broadcasting and a request-response paradigm (see figure 2). In a typical communication, the server broadcasts video content and a basic set of metadata to all clients via an unidirectional broadcast channel first. When a client wants to get more detailed information or wants to obtain additional content, a request is sent to the server. Both the client's request and the server's response are transmitted via the bidirectional interaction channel.

The server-side components of Vivid are described in the remaining sections of this chapter. The client-side architecture of Vivid is discussed in chapter 13.

12.2.4 System Output

The system output component includes three profiles: digital television, networked computers, and ubiquitous computing devices. In general, they employ different transmission channels and usually different content coding schemes. Thus, the chief tasks of the output component can be described as applying the appropriate encoding scheme, preparing content and metadata for delivery, and finally forwarding it to the transmission components.

Potentially, the output component might also have to compile and transmit the client-side application that presents metadata to the user and enables the user's queries. This, however, depends on the type of consumer device. This aspect is detailed in chapter 13.

Figure 3 illustrates the design of the output component of the generic system. The three distinct profiles are explained in the next few paragraphs.

DIGITAL TELEVISION. Since digital television standards rely on MPEG-2 as the encoding standard, the output module potentially has to transcode the content to MPEG-2. Besides, the metadata and digital rights information has to be transformed to a format that can be "understood" and handled by the client-side application program.

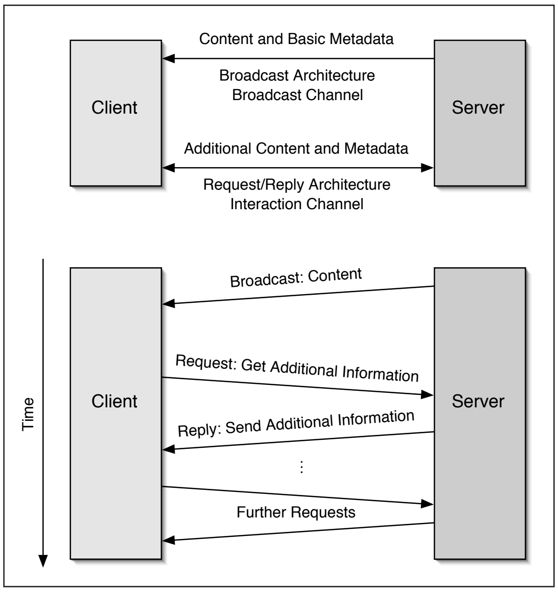


Figure 2: The system's communication architecture. Top: Communication channels between the client and the server. Bottom: Chronology of an exemplary client-server communication.

The converted data is packed into transport streams – audio- and video-streams for the content, and continuous data streams for metadata and digital rights. These streams are supplemented with additional DVB service information and a data carousel carrying the MHP application that will be executed on the consumer device (see sections 10.3.2 and 10.4).

Ideally, the client-side application should be available as soon as the user turns on the client device. If the application were transmitted in a data stream, for example, and the user switched on the client device after the application program had been delivered, there might be no way to access the additional functionality. Therefore a periodic transmission of the client-side application is necessary, which is preferably achieved with a data carousel.

NETWORKED COMPUTERS. Networked computers usually offer the greatest flexibility. Since it is relatively easy to extend the capabilities of a computer workstation by adding a software or hardware component, some of the complexity can move from the server to the client. Hence, it is basically not necessary to convert data before sending it to the client. The data can be transmitted in its "native" format. A special program residing on the client computer can decode and play back the received data.

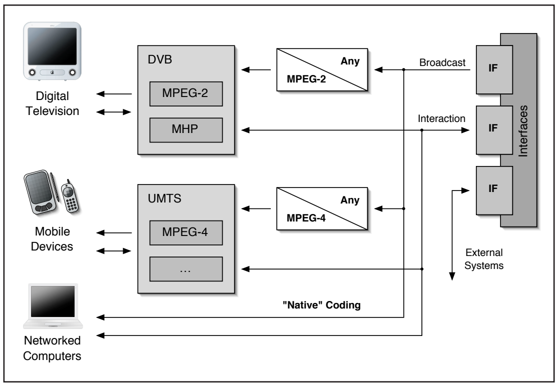


Figure 3: The system output component.

In order to reduce the amount of transmitted data, the client-side application program can be installed permanently on the client computer. In this case, a download is only necessary if a new version of the client application becomes available.

All video, audio, and data streams can be delivered via traditional computer networks such as IP-based Ethernet or ATM networks.

UBIQUITOUS COMPUTING. To an increasing extent, ubiquitous computing devices such as mobile phones and handheld computers are equipped with capabilities for receiving digital video content. Many vendors including Infineon, Sanyo, and Toshiba have chosen MPEG-4 for the implementation of video streaming features.

Infineon and Toshiba, for example, announced that they intend to produce mobile phone components for the new UMTS standard that offer hardware support for MPEG-4 (see [Infineon 2001]). In December 2002, Siemens presented a UMTS mobile phone with an MPEG-4 decoder and support for Java Wireless technology (see [Rutkowski 2002]). Another example is Sanyo, which has purchased an MPEG-4 license and plans to integrate it into PDAs. Also a software MPEG-4 player for Linux-based handheld computers is already available (see [Clarke 2001] and [Delbrouck 2002]).

The potential of current mobile devices is rather limited, though: with the upcoming UMTS standard, video streaming is only possible at data rates of up to 384 kbps, and at the moment the non-persistent memory of a mobile phone has a capacity of 2 MB or so.

The output module has to transcode the broadcast content to MPEG-4 and has to compile a data stream that contains metadata and digital rights. Since most mobile computing devices have limited resources such as small memory capacity, very small or no persistent storage, and small screen sizes, it might be necessary to "shape" the information before it is sent to the client. The content can be reduced in its quality – smaller resolution, less frames per second, lower color depth, poorer audio quality, mono sound only, etc. Besides, only the most important metadata can be transferred, and more detailed information is delivered on explicit requests by the user.

The audio, video, and data streams are transmitted using the MPEG-4 Delivery Multimedia Integration Framework (DMIF). It is a framework that permits delivery of MPEG-4 data on a variety of underlying network technologies. Among other transport systems, the GPRS and UMTS mobile standards are taken into account (e.g., [Perkis 2001] and [Fabri et al. 1999; Worrall et al. 2001]).

The client-side application is transmitted in an MPEG-4 data stream immediately after the connection between the service provider and the client device has been established.

12.2.5 System Input

The system's input data consists of the three required types of information: content, metadata, and digital rights. Typically the data stems from content producers and distributors, but it can also originate from internal sources such as existing databases.

In addition to this, there can also be a separate module that automatically generates new information from already existing data. This includes, for example, generating new metadata and movie previews from existing content. A schematic overview of the input component is depicted in figure 4.

The main task of the input component is to index and store content, metadata, and digital rights in the internal repositories in order to make it available to the system. Further functions can include conversion and filtering of incoming data.

CONTENT. Converting content from a given encoding scheme to the format the broadcasting environment uses, means essentially transcoding the data. Both software and hardware implementations exist for this operation. Popular products are, for example, MPEG-1 to MPEG-2 and MPEG-1 to MPEG-4 transcoders (e.g., [Ligos 2003; Taiga 2003]).

After transcoding the content, common object identification technologies are used to index the content. ISAN, the Content ID Forum's identifiers (cIDf), Digital Object Identifiers (DOI), and other standards can be utilized. See chapter 5 for details.

METADATA. When metadata is acquired from external sources, it might have to be converted. Very often this procedure can be fully automated and does not require user interaction. Examples are conversion from the Dublin Core metadata element set to MPEG-7 descriptors or from Open Archives Initiative metadata records to Dublin Core (e.g., [Hunter 2002]).

External sources can also provide metadata that is not directly usable. Screenplays, for instance, may contain very detailed metadata, but because of their unstructured nature, automatic extraction of information is often difficult. In a semi-automatic procedure, placenames, names of actors appearing in a scene, or the title of the corresponding sound track can be obtained. (The process is semi-automatic because ambiguities can occur, and user-input is needed to verify and refine the collected information.) If timecodes are available in the movie script, the extracted information can even be associated automatically with pieces of content.

Metadata from internal sources frequently stems from news archives, databases with program information, and teletext systems. Most probably all TV stations manage such repositories. The most elementary set of metadata includes the name of movie, names of some actors, the name of the director, the year in which it was released, available language, and the duration. Even in this case, it might be necessary to convert or rearrange the existing metadata records in order to optimize performance.

In addition to the presented functionality, the input component can include a module for generating new metadata. This generative process uses data that is stored in the content database to create new metadata. Whereas the methods described above focus on high-level features on semantic and bibliographic levels, this technique targets low-level features and structural information. Characteristics such as color histograms, texture, shape, pitch, and volume can be extracted automatically. The gernerated metadata can, for instance, be used for content-based retrieval.

DIGITAL RIGHTS. Since digital rights have not played a big role in analog television systems, many broadcasting organizations probably do not have dedicated repositories for this purpose, yet.

Thus, digital rights records might largely stem from external sources, mainly content distributors. In the future, digital rights information will possibly be obtainable for all newly produced content. This information will probably have to be converted and adapted to the system's rights system, rights expression language, and digital item declaration. Commercial products for an automated conversion of digital rights records are not yet available.

If external sources cannot provide intellectual property information for content (an old movie, for example) the corresponding data has to be created manually. Alternatively, all fields of digital rights records could be set to default values. The right to view the content may be granted, for instance, whereas extracting and copying are prohibited.

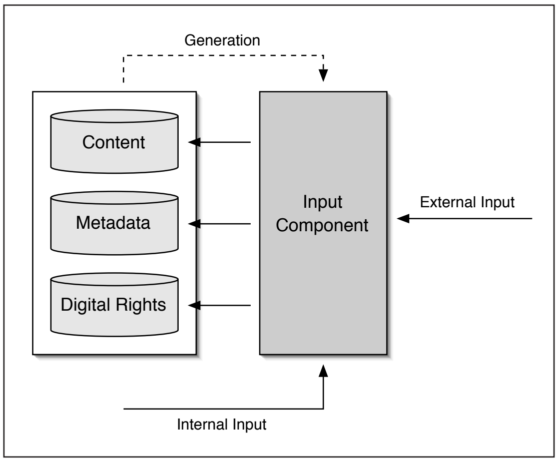


Figure 4: The system input component.

The advantage of the latter approach is that the action can be carried out fully automatedly without requiring any user interaction. However, the drawback is a certain inaccuracy that has to be accepted. Manual finishing touches can usually compensate for this.

12.2.6 The Core System

The system kernel consists of storage for content, metadata, and digital rights as well as the actual application and its interfaces to the output components. The major tasks of the core system are preparing the content and metadata for transport, handling users' requests, and communicating with external systems. Its central components are illustrated in figure 5.

APPLICATION. The main application is a collection of four modules for the following tasks: querying internal repositories, synchronizing content and metadata, communicating with external systems, and handling client requests.

The first module is for retrieving data from the system's repositories. It makes use of abstract functions such as `retrieve(content-type, identifier)` to access the archives.

Depending on the content type or the identifier (that includes an URI to the data source), a more specific function is instantiated. The actual function "knows" properties of the particular data source, the connection, and other details. The module is used when content is broadcast to all client devices or when a response to a user's request is assembled.

An autonomous module is utilized for synchronizing content with metadata and digital rights. This means that depending on the connection between metadata and objects or depending on timecodes (see section 12.2.2) small portions of metadata and digital rights are attached to certain temporal regions of the content.

This module is mainly used for broadcasting. Synchronization is usually not necessary when the user requests more detailed information on a scene.

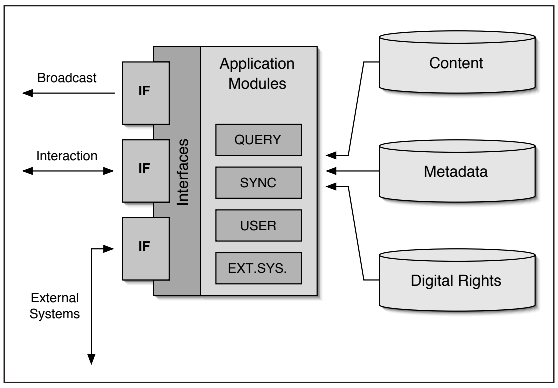


Figure 5: The system kernel with the application modules, its interfaces to the output component, and the storage system.

A separate module handles the communication with external systems. It queries external databases and contacts remote systems to obtain further information and supplementary content. However, it might also have to deal with incoming queries from external applications such as automatic program guide generators.

The module offers abstract functions such as queryExternalDB(db-type, db-name, query-string) to contact external resources. The type of the database, for instance, determines which concrete function is called. The particular function is adjusted to the characteristics of the external system. It can include special information on the underlying network infrastructure and authentication scheme, etc.

A fourth module is defined for the interaction with users. It manages incoming user requests and answers them by employing other modules of the application. When the request is received a query is assembled. It is either executed by the query module or passed on to an external system. After potentially synchronizing the resulting information with metadata, the complete record is returned to the user interaction module ("USER" in figure 6), and finally it is sent as a response back to the client device. The flow of information between the modules that are involved in answering a user's request is depicted in figure 6. (See also section 12.2.3 and figure 2 above.)

The module can not only be used for responding to users' requests. It can also be employed for the server-side storage of user preferences. These can be utilized to pro-actively disseminate specific content to users, suggest certain TV programs, etc.

An optional feature of the module is billing and accounting. Thus, things such as value added services, explicitly requested information, or the amount of transferred data can be charged to the user's account.

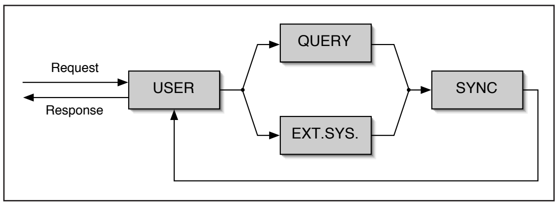


Figure 6: Flow of information among the modules involved in a user request.

INTERFACES. The system provides interfaces in the form of "ports" to communicate with output components and external systems. The interfaces are independent objects that are closely linked to the application. The advantage of this approach is that extensibility is made easier. Additionally, a major change of an output component does not require changes of the application itself but only of the interfaces.

The interface for the broadcasting channel is merely unidirectional. Output components can read the broadcast content that is supplied by the modules of the application from this port. The application as such does not need to know how many components are reading data from this interface or if the data is received successfully. From the application's point of view, the life cycle of broadcast information ends with sending the data to the corresponding interface.

The second interface that is responsible for the interaction channel is more sophisticated. It operates in a bidirectional way and can be compared to a "port listening daemon" in UNIX operating systems: the interface is permanently listening for incoming requests. Once a request from a client is received, a session between the client and the server is established. The interface passes the client's request on to the application modules that attempt to process it. The application's response (content, status or error messages, etc.) is forwarded to the interface that sends it to the client via the corresponding output component. After the requests and responses have been carried out, the session is terminated.

The third interface provides bidirectional ports to external systems. In this case, the interface can act both like a server and like a client. In client mode, it sends a request for a resource or a service to an external system and awaits a response. This request is generated by the application and may, in turn, originate from a consumer's request. The data from the external system is received by the interface and routed to the application.

In server mode, the interface listens for requests from external systems. The requests are passed on to other application modules that process them. After a response has been assembled, the interface sends it back to the external system.

Several instances of one interface can be created. This is true for all types of interfaces and makes it possible, for example, to use one interface for each link to an external system.

STORAGE. The storage architecture relies on a distributed system. There can be several different archives for content as well as various repositories for metadata and digital rights. This approach has several benefits and resolves some of the requirements (design goals) mentioned above:

- Different logical, physical, and geographic locations of the databases can be achieved. For large and geographically wide-spread organizations this can be quite common.
- It is easier to integrate existing resources and databases into the new system. Already established databases can be "linked" into the storage cluster.
- Various levels of access rights and security can be realized. The databases can be installed in different networks, which permits a (hierarchical) access model based on subnets.
- Performance issues can be addressed separately. The content database has to reach a much higher throughput as the rights database, for example. Therefore a different network (architecture) might be employed or a dedicated network connection can be set up.

As in every distributed system, a disadvantage is the cost of managing and maintaining the resources and relations among them. Furthermore an inexpedient organization can lead to interdependencies: if a significant resource becomes unavailable, the system might be left in a critical condition.

12.3 An MPEG-4-Based Architecture

This section illustrates the first of two specific architectures that are derived from the generic system overview. It relies entirely on MPEG standards: MPEG-4 is used as content coding technique, metadata is described with MPEG-7, and intellectual property information and digital rights are defined using MPEG-21.

This approach mainly targets networked computer workstations and ubiquitous computing devices. Digital television is also supported, but since it is based on MPEG-2, the content has to be transcoded prior to transmission.

The characteristics of this approach along with several products that can be used to implement the system are described in the next three sub-sections.

12.3.1 The Core System

The system kernel consists of dedicated databases for both digital media and largely text-based information. In addition to this, it contains the application modules and interfaces to the output component described in section 12.2.6. The structure of the core system is outlined in figure 7.

STORAGE. The MPEG-4 encoded binary content is retained in a special database that supports very large objects and permits the retrieval of them as both a stream and a single, coherent object. A database system that meets these requirements is, for example, Kasenna's Mediabase (formerly SGI Mediabase; see [Mediabase 2002; Kasenna 2002]). Among other content formats, the software can handle MPEG-1, -2, and -4 as well as MP3.

Every object in the database has a unique identifier that can be used to uniquely associate metadata and digital rights with the content.

MPEG-7 metadata is usually encoded using XML. Therefore an XML database is utilized to handle MPEG-7 data in the system.

Several products in this field are available. The Korean Information and Communications University, for instance, designed an MPEG-7 database that is based on the Microsoft SQL server (e.g., [Lee and Hyun 2002]). Among other descriptors, it supports low-level features such as color space descriptors. In contrast to this, eXist is a general purpose, open source, native XML database developed at Darmstadt University of Technology, Germany (see [Meier 2002; Meier 2003]). It is capable of managing and organizing any kind of XML formatted metadata. Hence, MPEG-7 metadata records are only one particular application.

MPEG-21 digital rights definitions are also based on XML technology. Thus, a database such as eXist can be employed to store and manage MPEG-21 rights records.

12.3.2 System Input

The system's input component transcodes and converts incoming data and stores it in the databases introduced above.

CONTENT. All incoming content has to be transcoded to MPEG-4. Several commercial software and hardware solutions are for this task are available. An example for such an application is the free, open source MPlayer/MEncoder software package (e.g., [Gereöffy 2002]). It can read a large number of different video and audio formats and is able to transcode them to MPEG-4 and other codecs.

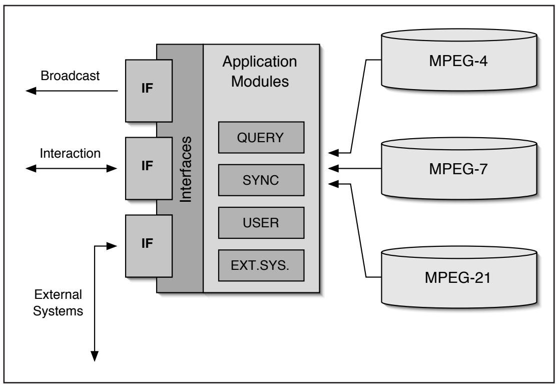


Figure 7: The MPEG-4-based system's core component.

When the content is added to the MPEG-4 database, a basic set of media specific metadata is collected: content size, duration, bit rate, etc. This information is automatically inserted into the metadata repository.

METADATA. Metadata from internal and external sources that is not MPEG-7 coded has to be converted. Although the mapping can be automated, this process is usually rather complex. A mapping from Dublin Core to MPEG-7, for instance, is detailed in [Hunter 2002].

Additionally, new metadata is generated automatically: low-level features such as color histograms, texture information, data about shape and motion; pitch, volume, etc. A number of research projects for automatic low-level MPEG-7 metadata generation exists. IMKA, for instance, is an application for content-based retrieval that is developed at Columbia University (see section 4.4.1 and [Benitez et al. 2001a,b; Benitez and Chang 2002a-c]). One of IMKA's core elements is a component for metadata extraction. Another example is the AGMA project carried out at the Fraunhofer Institute for Media Communications (see [Köhler et al. 2001]). Some functions of AGMA include face-recognition as well as speech- and music-recognition.

DIGITAL RIGHTS. Currently, no fully functional software for converting digital rights from one format to a different one is available. For certain rights systems it may not, even in the future, be possible to map rights records to a different system. This is true if the two standards rely on fundamentally different concepts or simply use incompatible entities to express the rights.

Therefore a rather pragmatic approach is chosen: if MPEG-21 coded intellectual property information is available, it is imported and stored in the rights database of the system. Otherwise default values are set: play back of video content is allowed, while other operations such as video extraction are prohibited; play back and extraction of audio material is allowed, whereas modification and duplication are not allowed; etc.

12.3.3 System Output

The system output component allows for three distinct output channels: for computer systems, MPEG-4 enabled mobile computing devices, and MPEG-2 based DTV equipment. The primary parts of the output component are displayed in figure 8.

NETWORKED COMPUTERS. For networked computers as client devices the system output is straightforward. Essentially, no transcoding or conversion is necessary because a software MPEG-4 decoder such as Apple's Quicktime Player ([Quicktime 2002]) can be installed on the target system. Depending on the implementation of the player application, the content and additional data have to be packaged into several streams or multiplexed into one single transport stream. The actual transport of the stream(s) is carried out by traditional computer networks.

Consumer appliances such as the KiSS DP500 presented in section 10.5 are also potential client devices. From a technical perspective they work in a similar way to computers: they receive the stream through the Ethernet connection, decode it with the built-in MPEG-4 decoder, process it with a CPU, and display it via the graphics system.

UBIQUITOUS COMPUTING. The second output channel aims at cell phones, PDAs, and other mobile computing equipment. UMTS-based networks and other technologies provide the required bandwidths that make it possible to transmit streaming media to those devices.

As mentioned in section 12.2.4, the bandwidth for UMTS transmissions is limited to about 384 kbps. Therefore the data has to be "shaped", viz., downsized before it is transported over the network. A possible division of the bandwidth reserves approximately 256 kbps for audio and video streams, and 48 kbps for an additional data stream. This schema utilizes 304 kbps of the total 384 kbps that are available (approximately 79 percent usage).

The value for an appropriate data rate for the MPEG-4 audio and video stream was determined in a very small test series. See Appendix B for details.

The MPlayer/MEncoder software package introduced in section 12.3.2 can be employed to “shape” the content: it can read the MPEG-4 video data, scale it down, reduce the number of frames per second, and finally produce the resulting new video stream. Similar operations can be applied to the audio stream to decrease the data rate.

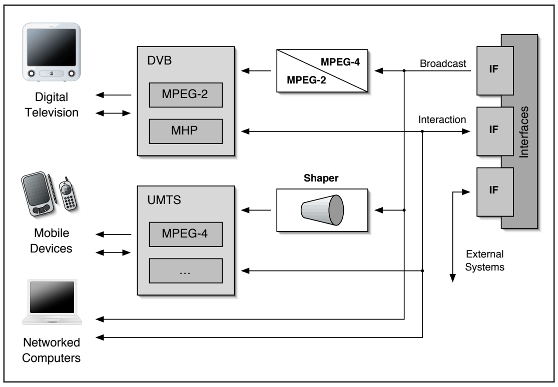


Figure 8: The output component of the MPEG-4-based system.

The metadata is transmitted in a compressed, binary format. An example is the BiM format that was developed for MPEG-7 (e.g., [Avaro and Salembier 2001; Wollborn 2001]). Moreover, merely an elementary set of metadata is delivered to mobile devices in order to use very little bandwidth. More detailed metadata is transmitted only on the user’s request.

DIGITAL TELEVISION. Since digital television standards rely on MPEG-2 as content encoding technology, the MPEG-4 data has to be transcoded. A specialized software product for this process has been presented by the Hongkong University of Science and Technology (see [Ahamad 2002]).

Several streams are created and transmitted using DVB: a video stream with about 5 Mbps, audio streams with approximately 250 kbps (stereo audio), and a stream with program specific information that consumes another 250 kbps.

12.4 An MPEG-2-Based Architecture

The second instance of the generic model described above aims at digital television as the target group. Mobile devices and networked computers are also supported, but the internal architecture of the system is adapted to television networks.

This section points out the dissimilarities compared to the generic system and the MPEG-4-based approach and explains some features of digital television systems.

12.4.1 The Core System

The basic design of the system kernel is naturally similar to the one of the generic system. The application itself and the interfaces to the output components are essentially the same. However, the storage architecture is rather different and quite complex. This fact is due to the variety of components that are employed in the production and distribution of television content.

The following few paragraphs describe the aspects of organizing and storing the digital content together with the associated metadata as well as digital rights and intellectual property information. An overview of the core system is given in figure 9.

CONTENT. Widely used content encoding standards in the TV production process include D1 (Sony) and D5 (Panasonic) (see [Martinek 1999]). Both are uncompressed digital formats producing streams with bit rates between 170 Mbps and 270 Mbps (e.g., [Utz 2000]). Another largely accepted professional digital format is Sony’s Betacam SX. It uses MPEG-2 interframe and intraframe compression with a compression ratio of 9:1, resulting in data rates of 18 Mbps (see [Sony 1999]).

The most recent successful and very promising development in this field is Sony’s MPEG IMX format (see [MPEG-IMX 2003]). The technology behind MPEG IMX is the MPEG-2 codec in its 4:2:2 Profile at Main Level (e.g., [Bagliani 2002a,b]). This configuration is considered as the most appropriate solution for the production of video content for digital television and DVDs – hence, the importance of this system and similar products.

MPEG IMX can produce bit streams at data rates of 50 Mbps. The material is recorded on tape media that are stored in large tape archives. Whenever a certain piece of content is needed it is retrieved from the tape archive – either automatically or manually.

However, the content can be accessed like a regular file on a hard disk, for example. This is achieved by introducing an abstract layer on top of the tape archive. In this layer, the content is either actually copied as a file to a filesystem or a database, or it is merely virtually represented as a file. When this “virtual file” is accessed, the content is directly retrieved from the tape (e.g., [Tahara and Gaggioni 2002]). So the application does not have to deal with the retrieval of the tapes, etc.

This basically means that the application modules, especially the query module, access the database via the interface that presents the MPEG-2 content as files.

METADATA. Although almost all TV channels already make use of metadata archives, hardly any companies employ uniform standards. In fact, many organizations have created their own tools and made up their own specifications. The metadata that is collected usually includes name and date of the program, names of scriptwriters, editors, cameramen, and cutters, etc. Basically, the metadata resembles the information in the Dublin Core metadata element set.

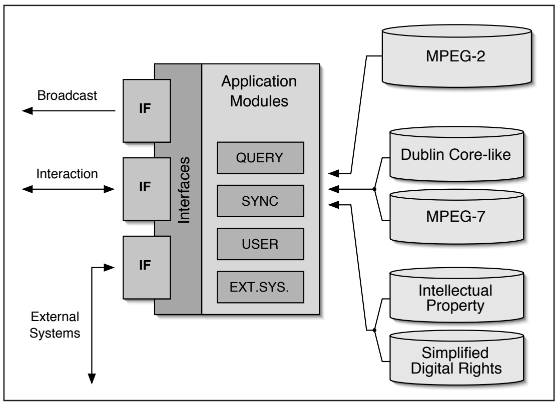


Figure 9: The system kernel of the MPEG-2-based system.

Although the bibliographic metadata is valuable, it is not sufficient for this kind of application. Therefore an extra database is added to the system. It is MPEG-7-based and supplements the existing information with detailed metadata stemming from movie scripts and similar sources. Moreover, references to related material or external resources can be included.

The MPEG-7 metadata can largely be generated automatically, with little or no human interaction (see section 12.2.5). The indexing technique for both MPEG-7 and Dublin Core-like metadata (and also for digital rights) in this approach is timecode-based as described in section 12.2.2.

DIGITAL RIGHTS. At the moment, most TV channels maintain databases with rudimentary intellectual property information. It is used to find out who actually owns the content (i.e., the copyright) who the distributor is, etc. These records are very often stored in (proprietary) database systems and do not follow any major standards.

Up to now, it has not been necessary for TV stations to retain more information than simple intellectual property records. The reason is that TV stations themselves have not been redistributing content in a way that requires more detailed specifications of the intellectual property data. However, the proposed system does not only need intellectual property information but also digital rights. This means that not only the information about who owns the content is required but also what the consumer is allowed to do with the content.

Therefore a new, separate database is employed for this purpose. It keeps at least a simplified set of digital rights that state if content may be copied, extracted, reproduced, etc. The additional cost of maintenance that is caused by this database is relatively low because:

- the information input is largely automated; and
- database queries are generated and handled automatically by the "QUERY" and "SYNC" modules of the core application.

12.4.2 System Output

As in the generic system, the output component is responsible for assembling the data and making it ready for delivery. The flow of information and the most important elements of the output component are shown in figure 10.

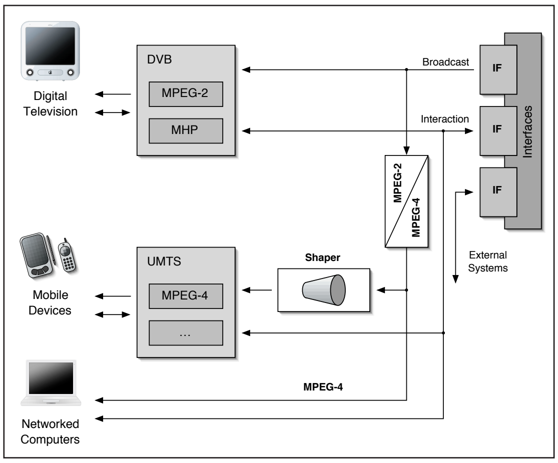


Figure 10: The output component of the MPEG-2-based system.

DIGITAL TELEVISION. The content provided by the core system is at production quality: MPEG-2 encoded with the 4:2:2 Profile at Main Level. For distribution via digital television networks it has to be downscaled and transformed to the Main Profile (4:2:0) at Main Level.

The resulting audio and video streams consume approximately between 4 and 6 Mbps. A data stream carrying the binary coded metadata and digital rights is synchronized with these streams.

NETWORKED COMPUTERS. The MPEG-2 data made available by the system is transcoded to MPEG-4, the preferred format for content delivery over computer- and mobile networks. Together with metadata and digital rights records, the high-quality MPEG-4 stream is transmitted over conventional computer networks. On the client-side the data is decoded and displayed by a software MPEG-4 player.

MOBILE DEVICES. The MPEG-4 data that is generated for the output channel designated to computer networks can be seen as a precursor of the content stream for mobile devices. This high-bandwidth MPEG-4 stream is downsized and degraded in quality. (That process is depicted by the "shaping" element in figure 10.)

The lower-bandwidth MPEG-4 audio- and video streams are transmitted together with supplementary metadata, digital rights, and intellectual property information via an UMTS-based network. Details of this scenario are described in section 12.3.3 above.

12.5 Conclusion

This chapter has outlined the architecture of a generic system that can be used in a wide variety of production environments from digital television to video streaming servers. The component-structure of the system makes it possible to reuse elements of other, already existing systems. This also allows for future extensions of the system – further internal resources, new content acquisition methods, additional distribution channels, etc.

Based on the generic system, two concrete approaches have been presented. The first concept makes use of state-of-the-art technologies such as MPEG-4 for content coding, MPEG-7 for the description of metadata, and MPEG-21 for the definition of intellectual property information and digital rights. Basically, this approach is not extremely complicated because a completely new system is defined, and existing components do not have to be taken into consideration.

The second system focuses on a real-world digital television environment. With MPEG IMX it employs one of the latest digital video systems. However, there are also some rather inconvenient conditions such as non-standardized metadata initiatives and sometimes non-existent digital rights information. These missing or incomplete parts of the system have to be created and integrated into the system in a way that keeps both cost and the level of maintenance low.

Hence the probably most critical step in implementing the proposed system can be described as bringing together existing elements and newly created parts in order to form an advanced system for digital video broadcasting.

The two examples indicate that the chosen architecture for the generic system is quite sensible, in that it is flexible enough to be adapted to the various demands of the different environments and backgrounds.

13 Advanced Player Applications

13.1 Introduction

In Vivid, the device to reproduce the encoded digital media runs a piece of software, the player application, that decodes, interprets, and presents the content and its extensions. Depending on the degree of flexibility imposed on the system and the availability of a return channel and a connection to the internet, two major categories of players can be distinguished: a more or less standardized, simple version with a limited set of core features, and on the other hand an open, potentially complex version with more sophisticated features. Many nuances in between those two systems may exist, of course.

This chapter explains the different architectures of applications for networked computer systems, mobile computing devices, and digital television equipment. Subsequently, two different approaches to a player application and some of the key aspects are described. An extensible version of the player software is characterized thoroughly.

13.2 Application Architectures

Since there is a wide range of client devices, from digital television to personal computers and mobile phones, the particular underlying architectures and different capabilities have to be considered. Therefore the key requirements for the design of the player are listed in the next sub-section. After outlining the overview of a general system that meets these demands, three specific approaches for networked computers, mobile devices, and digital television equipment are presented.

13.2.1 Design Goals

The player software and all its features should be available on a variety of hardware- and software-platforms. On the other hand, it should be made as easy as possible for programmers to implement and update the application. These design goals are expressed in the next few paragraphs.

ADAPTABILITY. It should be relatively uncomplicated to adapt the player application to different hardware architectures and software platforms. (In pure software environments this requirement is usually called "portability".)

EXTENSIBILITY. There should be an uncomplicated way of adding new features and functions to the software. Also third-party developers should be able to add new features to the application.

SMALL FOOTPRINT. Some of the potential client platforms offer limited capacities. Therefore it is necessary to create a lightweight application that does not consume much memory or computing power. Furthermore the application should be small in size to keep transmission times low.

MAINTENANCE. The cost of maintaining the software should be as low as possible. Ideally, the player application is written once and can be used without any modifications on all client platforms.

13.2.2 Architectural Overview

The system is designed to be modular and extensible. The architecture enables the software to be ported to a wide variety of different devices and software platforms without the need to redesign the entire system. Therefore the software consists of an application kernel that can be utilized on all platforms. It is extended by several interfaces to external, platform-specific system resources. The core system includes modules for the following the essential tasks:

- decoding metadata and digital rights;
- synchronizing and "merging" metadata and digital rights with content;
- preparing a visual representation of the information and displaying it to the consumer;
- providing input channels for user interaction;

- assembling queries;
- executing and answering queries (locally);
- sending data to and receiving data from the service provider (sending a request and receiving the response, for example); and for
- extracting objects and saving them to the internal storage space.

Largely independent from these core modules a number of interfaces exist (depicted by the "IF" elements in figure 1). They provide access to the underlying hardware and to platform specific features. The set of interfaces includes:

- an interface to the transmission and decoder hardware for reading the decoded content and received data;
- an interface to the display hardware to present content and supplementary information;
- an interface to the input hardware in order to read the user's interaction;
- an interface to the return channel for sending data back to the service provider; and
- an interface to the storage space of the device (if available).

The advantage of this approach is that the client software components of Vivid do not have to be aware of any particular details of the underlying hardware. The application program only assumes that there is, for example, a display and that it can present information on this display. The specific interface contains the precise properties of the device: the type of display, whether it supports colors or not, etc.

APPLICATION LAYERS. The basic architecture of the application can be seen as a layered model like illustrated in figure 1. The central component contains the application core modules. The application receives the incoming (broadcast) data from the corresponding interface to the transport and decoder component. After the application processes the data, it is forwarded to the display interfaces that present the content together with the supplementary information to the user.

Figure 1 merely shows an abstract model for the dissemination and processing of broadcast content. However, a similar concept is employed for user interaction. The user issues a request via an appropriate input device. Then, the application reads the user's request from the corresponding input interface and compiles a query. In case all necessary resources are available, the query is answered by the application itself (local computation). Otherwise the query is sent back to the service provider via the return channel interface.

So while the dissemination process is "bottom up" (from the transport hardware via the application to the display hardware), the user interaction is "top down": from the input device through the application to the transmission hardware. (In figure 1, the display hardware would have to be replaced with input hardware, and the direction of the arrows would have to be reversed.)

PROGRAMMING ENVIRONMENT. Although almost any programming language could be used to implement the player application, the preferred software development platform is Java. The reason is that software development on both digital television equipment and mobile devices (cell phones, handheld computers) makes use of Java technology. Personal computers are, again, flexible and can, of course, also handle a Java-based player application.

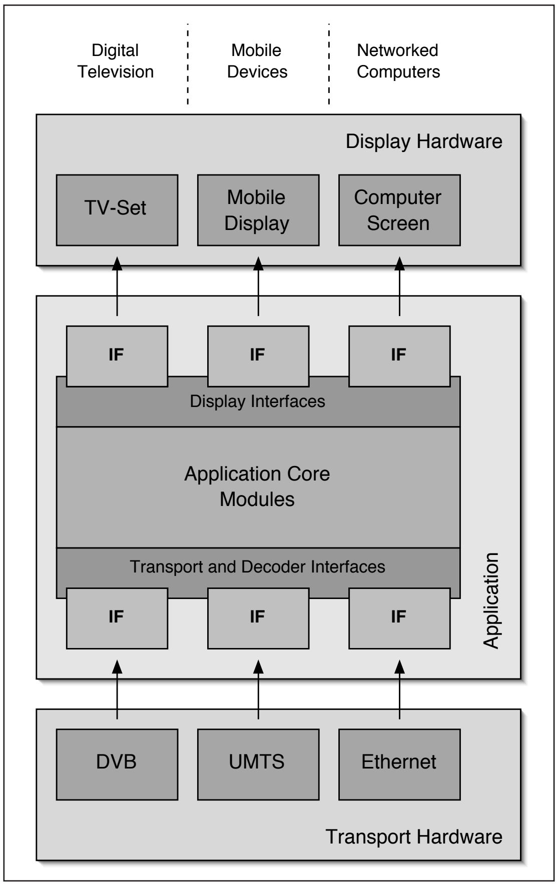


Figure 1: Relation of the several layers of the application and the involved hardware components in the process of receiving and displaying data.

Thus, the application for all three different target groups is based on Java, which keeps the cost of managing and maintaining the software comparatively low, and facilitates the reuse program components at the same time. This means that the core modules for querying or decoding metadata, for example, are identical on all systems, and only the platform specific elements such as interfaces have to be adapted.

13.2.2 Networked Computers

The preferred content encoding technique for networked computer workstations is MPEG-4 because high-quality video can be provided at relatively low data rates. Moreover, ubiquitous computing devices can also make use of MPEG-4 encoded data (see below).

Computers can use a software decoder to obtain the content from the incoming media streams. A wide variety of programs for this purpose is available. One particular example is IBM's MPEG-4 decoder that is implemented using the Java Media Framework (see [IBM 2001; IBM 2002]).

The advantage of this approach is that this MPEG-4 decoder can be part of a bigger Java application. This means that the decoder component of the client-side system, that is part of the lowest layer in the model in figure 1, becomes a module of the player application, which is illustrated in figure 2.

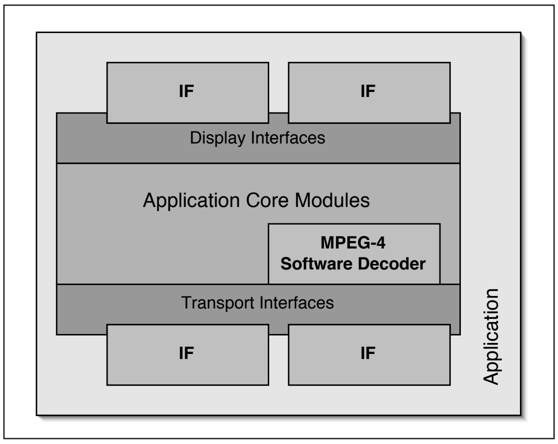


Figure 2: The core application modules with an integrated software MPEG-4 decoder.

Both the modules of the application kernel and the interfaces to the computer hardware are to be implemented with the traditional Java software development kit. The transport hardware, for example, can be accessed using the java.net framework, the display hardware with java.awt or Swing packages, and the content decoder is accessible through IBM's MPEG-4 Java software.

13.2.3 Ubiquitous Computing Devices

To an increasing extent, mobile devices have built-in hardware MPEG-4 decoders and provide a Java Virtual Machine (see section 12.2.4). The device receives the media and data streams, demultiplexes them and passes them on to the MPEG-4 decoder. Now the application can access the decoded data via the interface to the transport and decoder hardware.

The core application is realized using standard Java components from packages such as java.lang or java.io. The application's platform specific parts, however, are implemented with Java Wireless technology (see [Java-WTK 2002; Java-WTK 2003; Java-WDS 2003]). The toolkit offers access to components such as the transport- and display-hardware. In the model in figure 1, the interfaces correspond to Java Wireless technology, whereas the application core modules are "plain Java".

13.2.4 Digital Television

The digital television equipment receives the DVB multiplex that contains MPEG-2 audio and video streams as well as other digital data. After demultiplexing the various streams, a hardware MPEG-2 decoder produces the audio and video content from the respective streams. The raw data from the data stream is directly passed on to the main application in order to retrieve the encapsulated metadata.

The development of the client-side application for digital television devices is based on the Multimedia Home Platform (MHP, see section 10.4). This means that the application kernel can, again, be implemented with common Java components. In contrast to this, the interfaces to the display and the underlying DVB hardware are implemented with the frameworks provided by MHP.

13.3 Standardized Player

A standardized player aims at all sorts of devices with limited networking capabilities and interaction channels. On the other hand, this application is also well-suited for people who want or need only basic functionality.

The player provides a well chosen set of elementary functions that can be used for a large number of different types of media and objects. Furthermore it should be capable of answering the users' most frequently asked questions.

Figure 3 shows the prototype of a user interface of such an application on a home computer. At the first glance, the player (without the "drawer" on the right-hand side) strongly resembles a traditional media player software with an area for displaying the content and the five classic navigation elements: play/pause, fast forward, rewind, go to the beginning, and go to the end.

This can be seen as a seamless transition for the users who won't encounter difficulties using the new application because there is no obvious visual difference to the interface they are used to. The new features are accessible through the buttons "Information" and "Store Object".

13.3.1 Object Information

Detailed information on a certain object on the screen can be obtained at any time. The user can do this by pausing the playback, selecting the particular object with the pointer device, and pressing the "Information" button.

Figure 3: Prototype user interface of a player application for Vivid. Top: if no object is selected and the user presses the "Information" button, a list of object in the scene is shown. Bottom: information about a particular object (the actress) of the displayed content. The picture of Ellen Feiss was taken from [Feiss 2002].

This, of course, only works for visual content including the background image or a moving object. However, for non-visual content such as speech or music this approach is inappropriate. In this case, the user pauses the scene and presses the "Information" button (without selecting an object) to get information on the scene. All elements that occur in this scene are listed, and now also non-visual objects can be chosen. An example for a list of objects in a scene is shown in figure 3 (top).

The window for displaying information on an object is subdivided into three parts: the object information, the content description, and associated digital rights. The object information is provided by the content decoder and is the data that is related to the characteristics of the content. It describes the technical specifics of the media and is basically the kind of information that is available in traditional media players. Examples are an object identifier, the content type (still image, moving region, audio, etc.), content encoding (MPEG-4, JPEG, MP3, AAC, …), bit rate, data size, duration in seconds, and the transport medium.

The content description stems from the metadata attached to the object. Although the description could essentially contain the full range of (MPEG-7) descriptors, from low-level features to bibliographic information, the simplified version of the player only processes those entries that are considered to be of most interest for the user. This data includes, for instance, the real-world name of the object, the date and place where it was captured, the date when it was published, and the name of the content creator. Presumably the bibliographic elements of the MPEG-7 description are the most relevant ones for the majority of users.

The third subsection contains the digital rights record. Again, not all information that is available is actually displayed. Only a few aspects are presented to the user. The ownership of the content, the copyright, copy-protection, and permissions to store or print the object are only a few exemplifications.

13.3.2 An Example

Figure 3 (bottom) illustrates an example scene with the information window for one selected object. The scene consists of four MPEG-4 objects: a still background image, a motion picture in the foreground, and an audio track for the background music as well as one for speech. The user has already paused the playback and selected an object. The selection is highlighted and the information is displayed at the right side of the window.

The object is an MPEG-4 encoded video stream. The MPEG-7 descriptors state that the name of the actress is Ellen Feiss, and that the video was published in July 2002 by Apple Computer. The digital rights tell the users that they are allowed to view the content for an unlimited number of times. However, they are not allowed to extract or copy it.

Since the user is not allowed to save the object to disk, the "Store Object" button is not enabled in this example.

13.3.3 Prospective Use of the Technology

As shown in the example above, the user can easily have the question *"Who is this actress?"* answered by using the advanced features of the application. The actress's name can be found in the content description, thus an answer can be provided within seconds.

Another feasible operation is the acquisition of the sound track of a movie. The song is available as a separate MPEG-4 object, the MPEG-7 content description contains the name of the artist and the title of the song, and the MPEG-21 data says, for example, that the audio material can be freely downloaded to an external storage medium and that it may be played back for an unlimited number of times. Since a copy-protection mechanism is employed, it cannot be duplicated, though.

Although the user already has the ability to have some questions answered and makes use of some new features such as partially downloading content, one of the initial questions is still unresolved: *"Where have I seen this actress before?"*

If the device for reproducing the content is used in a networked environment (e.g., a home computer with an internet connection or potentially also cable TV) even this request can be fulfilled relatively easily. Since the name of the actress is available, a dedicated movie database such as the Internet Movie Database (IMDb, [IMDb 2002]) can be queried. IMDb is only an example – other resources such as research databases or an archive with newspaper articles might also be useful.

This section has only provided three examples of how the advanced features of the simple player might be used. There are certainly many more, but they have one thing in common: they make the accessibility of information and the navigation in digital media easier – even for the inexperienced user.

13.4 Open, Extensible Player

The second type of the advanced media player is meant to be very flexible and extensible. The intent is to use this application in on devices such as home computers or internet terminals. Rather experienced users with a clear idea of a distinct feature they are searching for might want to use this tool.

As already indicated in the introduction, a number of variants between the standardized player and the extensible one may exist. It is, of course, also possible to integrate only a few selected functions from the extensible version in the simple player.

13.4.1 Comparison with the Simple Player

This second variant of the player provides all features of the basic version. Therefore it is fully backward compatible with the simple player. Although there is a certain similarity between the two applications, the user interface and the "look and feel" of the programs can be quite different.

This situation is best illustrated with an example. The CD-player of a home entertainment system and a software CD-player on a computer have the same purpose, and therefore the basic features need to be the same. Even if the user interface of the software is completely different from its counterpart and it has far more features, the basic set of functions is still the same.

Both the basic and the more sophisticated player use the same set of data, but they process different parts of it and present different results to the user. The basic player, for example, only displays high-level, bibliographic metadata. The sophisticated player, however, does not only use high-level descriptions but also low-level features to enable a whole range of new functions. This is achieved by using a modular extension system.

13.4.2 Examples for Extensions

Extensions can use any kind of material that is accessible in the transmitted stream. Thus, it can be related to the content itself, to the (MPEG-7) descriptors, or even to (MPEG-21 coded) digital rights information.

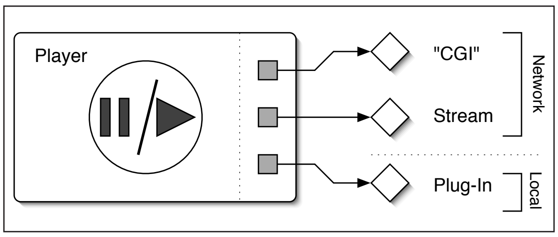


Figure 4: The player provides "hooks" to tie in different kinds of extensions: plug-ins, CGIs, and extensions transmitted within the stream.

A rather simplistic extension is a module that links all kinds of metadata to an external database. The external source could be an archive of "media and people". When the user selects the name of an actor, all information about the actor is retrieved from the archive; when the user selects a place, information about the place is displayed; etc.

A more elaborate module could use the low-level features of MPEG-7 descriptions in order to find similar objects (content-based retrieval). Users could run a query for objects of a similar shape, for regions with a similar texture or similar colors (e.g., [Chen et al. 1999]). Another task could be *"Find a similar melody"* (e.g., [Arentz 2001; Herrera-Boyer 2002]). These procedures include techniques such as similarity recognition and pattern matching.

Yet another extension could create a semantic object map. This is a graphic representation of spatial, temporal, and spatio-temporal relations among objects in different scenes. If two actors appear in a scene at the same time, for instance, there is obviously a relationship between. This relationship can be graphically depicted as semantic map (e.g., [Boyack et al. 2001; Chen 2001]).

A probably useful tool is a "personal movie guide", something like a (structured) scrapbook. Users can retain their favorite scenes of movies (or references to them) in the scrapbook. They can add photos of actors, make annotations, copy descriptions, insert short sound-clips, etc.

An extension module that collects user data, for example, can enable a whole range of new applications such as pro-active content dissemination, automatic downloading of certain content-related information (based on user preferences), etc. However, these techniques raise privacy issues that should not be neglected.

The techniques described above are only a few examples, and there are, of course, many more innovative extensions.

13.4.3 Technical Aspects of Extensibility

There are two different kinds of extensions: optional add-ons and required plug-ins. An add-on provides, as the name suggests, additional features or new functions. These functions are available to all objects of a certain type. The content itself does not have to be aware of the add-on.

The "personal movie guide" is such an add-on. If it is available, it is a practical tool that can add new features. However, it is not required to view the actual content, and the content does not necessarily have to be aware of the existence of the "personal movie guide".

Opposed to that, plug-ins are needed to enable certain features. If a required plug-in is missing, a function might not be available. A module for creating the semantic object maps is an example. If certain sections of the content rely on this module, the corresponding features are disabled if the module is not available. In this case, the content has obviously to be aware that a plug-in is required.

Thus, two approaches can be identified. There can be add-ons to analyze the content. If the use of add-on is appropriate, it is enabled. Ideally, the player indicates to the user which add-ons are available in each scene or for every object. Thus, in this approach the extension is the "active" part.

Alternatively, when required plug-ins are used, the content carries particular information that further features that extend the basic functionality are available. If the user wants to access one of these features, the corresponding extension has to be loaded, and then the operation is carried out. In this case, the content takes the "active" role.

13.4.4 Implementation of the Extensions

There are several approaches to implement the described extensions. The aim is to provide the essential functionality of the basic player in a static program. Consequently, it makes "hooks" available where extensions can be tied in.

There are three different methods to extend the functionality: local extensions via plug-ins, remote extensions using CGI-like programs, and extensions delivered in the stream (see figure 4).

PLUG-INS. Plug-ins are external modules to enhance the range of features. This technique is available in many other programs such as web-browsers. Plug-ins reside on a local storage device, which can significantly improve the performance. When the users selects a function that requires the plug-in, it is loaded into memory, all required parameters are passed on to the module, and the operation is carried out.

A plug-in could offer standardized functionality for a certain field of application. That way, many different content providers can access the same set of standardized functions provided by one plug-in. A general disadvantage of plug-ins is that some users are not able to or simply do not want to install plug-ins.

IN-STREAM TRANSMISSION. An approach that is similar to the delivery of the player application itself is the utilization of in-stream transmission. The extension is either transmitted in the same stream as the digital content or it is transmitted in parallel in a separate, dedicated out-of-band data stream. After the extension has been downloaded, it is embedded into the player and made available to the user.

This approach basically solves the problem associated with plug-ins: the user does not have to install a plug-in because the software extension is installed automatically. However, this approach is more difficult to implement.

CGI. Common Gateway Interfaces (CGIs) are a technology stemming from HTTP-servers. A CGI defines an interface between the HTTP-server and an external program that is called by the server. In general, the concept is typical for a request-and-response architecture.

The CGI-approach can also be used for the extension of the player application. When the user selects an external function, the necessary parameters are transmitted to a specified URL. The remote system processes the data and sends the results back to the player that displays the received information.

This technique reflects a traditional client-server system with its disadvantages: although the calculation on the remote system might be completed in the fraction of a second, transmitting the request and receiving the result can take much longer on a slow network. Also large amounts of data might have to be sent across the network.

13.5 Conclusion

This chapter has presented two prototypes for player applications that could be used to implement the notion of an advanced digital video broadcasting system. The basic variant resembles a traditional media player, but still offers a set of smart features that are easily accessible.

In addition to this, another version with more sophisticated features for professional users has been outlined. It is rather aimed at networked computer workstations (home and office computers, internet terminals, etc.) and allows for flexibility and uncomplicated extensibility.

From a technical point of view, different possibilities to extend the core system have been explained, and the functionality of several possible extensions has been drafted. The sketch of the prototype demonstrates that the interface can be designed in a way that it can be used easily, even by non experienced users.

Conclusion to the Thesis

Summary

The first part of this thesis provided an introduction to knowledge management and presented several techniques that are currently employed in knowledge management, including metadata description with MPEG-7 and digital rights management.

Part 2 focused on active documents, a concept that originates from and is commonly employed in knowledge management systems. In chapter 8, the use of the technique was extended from classical text-based environments to all types of multimedia objects. Thus, text-based active documents are transformed to active multimedia documents.

An example for the application of active multimedia documents resulted in the proposal of ADIME, an innovative addition to conventional e-learning environments in the field of medical education. The presentation of ADIME in chapter 9 showed how active multimedia documents can enhance the learner's experience in a media-rich learning environment that includes objects such as images and video clips.

Part 3 introduced the characteristics of digital video broadcasting and briefly addressed software development for digital television equipment. In chapter 11, these basic foundations were used to describe VIVID, a proposed system for enhanced interactivity in video broadcasting environments. The fundamental idea of the system was presented together with a number of usage scenarios. Additionally, the role of the different types of information in VIVID was detailed.

The design of a system architecture for VIVID was described in detail in chapter 12. A generic architecture that contains the major components of the system was introduced. It consists of a system core, an input component, and an output component. Based on the generic system, two concrete instances were detailed: an MPEG-4-based approach with networked computers and mobile devices as key target group, and an MPEG-2-based model that mainly aims at digital television. Finally, the last chapter presented two versions of a client-side player application that is capable of displaying the information provided by VIVID on consumer devices such as a digital television sets or computer screens.

Further Research

Although this thesis introduced extensions to active documents and discussed the idea of active video broadcasting in detail, there is room for further research. Many aspects are simply not within the scope of the thesis.

VIVID, for instance, requires further refinement in many parts. For the basic concept, a survey is necessary to find out which questions are frequently asked by users and which questions are most important to them.

Also the source of metadata needs to be made clear: although content providers usually do maintain metadata repositories, their formats are often incompatible. In addition to this, characteristics such as the utilitzed metadata standard or the level of detail for different archives vary.

For an implementation of VIVID, the architecture of the proposed system has to be refined: the structure of the core system's databases needs to be detailed, the core modules have to be defined on a much more detailed level, etc. The probably most challenging task, though, is the precise specification of the output component and the implementation of the client-side software. In this case, many different standards and technologies are involved, which makes an implementation very complex. (The digital television output module potentially has to deal with at least four standards: MPEG-2 for the content, DVB as a "wrapper", MHP as a software development platform, and a separate standard for transmission.)

Thus, for VIVID still some research can be done, and it can even be improved in some facets. Also, the idea of ADIME should be pursued, and aspects such as the user interface, the iconic approach for active documents, or the integration in existing systems can be refined. Furthermore, a prototype for ADIME should be produced.

Conclusion

This thesis demonstrated how methods such as active documents can be employed in domains that have received little attention from knowledge management initiatives so far. The complexity of VIVID's architecture probably shows why knowledge management has not yet been applied in these areas: many interacting components, a whole range of standards and requirements that have to be considered, and the description of multimedia content are only a few characteristic features that make the design of applications difficult.

With ADIME and VIVID, the concept and architecture of two innovative systems were proposed. However, further research needs to be done to enhance the described ideas and illustrated methods. Moreover, it is necessary to refine the presented systems in order to be able to produce working prototypes.

Appendices

Convergence of Television and the Internet

A

A.1 Introduction

This chapter gives a brief overview of my ideas about the convergence of digital television and the internet. First, the role of the internet and television in the people's lives and the capabilities of the two technologies is described. Building on these findings, reasons for me to believe that the two areas are actually converging are presented. Furthermore a scenario for the future of current client devices is outlined.

A.2 Capabilities and the Role of Television and the Internet

At the moment television is by far the most popular way for people to stay informed. One reason is that this kind of information is readily accessible, and that TV is a medium that is easy to use. Basically, people do not have to do any "think work" when they are watching TV. They just turn it on and watch.

Each TV program has been more or less carefully produced by someone else. *Ideally,* the producer (or director) of the program has done the research on a topic, selected and "digested" the most important information, and assembled a well-rounded, new piece of content.

A typical example is a documentary, where the director brings together information from several sources and extracts what he or she thinks are the most significant facts. As a result, the consumer should get a solid, but usually not extremely detailed overview of a topic.

The internet, on the other hand, offers much more information, a greater variety, many different sources and potentially more details. However, it is also much more complicated for users to find suitable information. Moreover, it is necessary to obtain information from more than one source in order to get an objective view of a matter. For many consumers this way of information retrieval is simply too complicated and time consuming.

Therefore I dare to predict that the vast majority of consumers will also in the (near) future watch conventional (but then digital) TV to get a brief overview on a certain topic. It should be remarked, though, that "watching TV" is not restricted to the TV set as consumer device, and "getting information from the internet" is not confined to computers. The consumer will be able to watch a documentary and request information from the internet on the same appliance – be it the TV set, a computer, or a ubiquitous computing device.

If users want to know more about something they see on TV, they will get the information from the internet. An example for such an application is Vivid, the system proposed in Part 3 of this thesis.

Thus, television will still be the main source of information for most people, and the internet will be more like an add-on to traditional TV services.

A.2.1 View of the Relationship

The relationship between television and internet can be described in a more graphical way. There is a horizontal and a vertical approach.

Television channels provide a broad view in that they offer many programs that supply only more or less superficial information. The programs depict a little bit of everything without going too much into detail. They present overviews and selected articles, i.e., they are providing an entrance point to a topic. This can be understood as a horizontal way of accessing information.

In contrast to this, the internet as a supplementary resource can be seen as vertical approach. It provides in-depth information and much more facts than the rather general TV program. Hence, the detail that simply cannot be provided by conventional television is offered through a close link to the internet as a vast information archive.

A.3 Converging Technologies

As the examples above show from a content perspective, the internet and television as content providing services are converging. Also from a technical perspective the two technologies are converging: digital television (DTV) is based on MPEG streams that have successfully been used on computers for several years. Some DTV standards provide facilities to display HTML pages directly on the TV screen, and even set aside for Java and ECMA-Script, a generalized variant of Javascript.

Basically, the same is true of mobile devices: many of them are ready for MPEG-4 encoded media, and internet access and the support for Java technology are also among standard features.

These core technologies – MPEG streams, access to the internet and the WWW, and integrated support for Java technology – can be used by *many* different client devices including integrated digital television sets and set-top boxes, computers, mobile phones, and handheld computers. And as mobile devices become more powerful and computer and mobile networks provide higher bandwidths it becomes more and more feasible to watch movies and news broadcasts on a mobile phone or on a networked computer. On the other hand, it is possible to access even media-rich content from the internet on handheld computers and digital television equipment.

A.4 Diverging Technologies

Although technologies – *content distribution* technologies – are converging, there will at least in the near future be a wide variety of ways in which these technologies are manifested. This means that although different devices have the same capablities, they will still be distinct devices.

The reason is that each device still has a “main task” or a specific purpose. Mobile phones, for instance, will still be mainly used for talking to other people, and functions such as internet access and video streaming will be extensions to the basic functionality.

Another example is digital television. It will still have one main function: watching movies, documentaries, news, etc. People who are interested in a certain topic will have the ability to use it in order to retrieve additional information from the internet or use other functions such as e-mail or e-commerce applications. However, it will still be used like conventional TV by the majority of consumers.

Apart from the consumers’ preferences, there are also different physical requirements. Devices for ubiquitous computing, for example, have to be small, light, and easy to transport. Hence, the devices as such and their displays are small. Some of the devices are even so small that they are built into other things such as glasses (e.g., [Maurer and Oliver 2003]). So the basic tendency could be summed up as “smaller is better”.

In contrast to this, for a TV set mobility is not important at all. Usually it is installed permanently in a house, and there is no need to move it. Therefore other characteristics are significant: a big screen, a clear picture, the sound, etc. In this case, the trend could be described as “bigger is better”.

To put it a different way, the range of different client devices will most likely have very similar capabilities, but each kind of device will still focus on certain aspects. Therefore even in the (near) future different appliances with distinct purposes and specific shapes will exist.

A.5 Conclusion

The application areas of the internet and of digital television are going to merge. This can be explained with the similar technical capabilities of the corresponding consumer devices. This situation has also been discussed by many other authors such as [Lennon and Maurer 1994] or [Rose et al. 1999].

However, this development will not prevent computers, television sets, mobile phones, and handheld computers from existing as separate devices. Most people, I believe, will still own a television set for watching TV, a computer for using the internet, a mobile phone to call people, and a PDA to organize things.

MPEG-4 Video Compression Test

B

B.1 Introduction

The system architecture for Vivid described in chapter 12 defines an MPEG-4-based approach that can be used to deliver digital audio and video content to mobile devices such as cell phones or handheld computers.

This chapter documents a test series that was carried out in order to determine an appropriate MPEG-4 compression level and data rate for use with mobile devices. First, the test environment is described. Then the test results are discussed briefly, and one particular configuration is recommended.

B.2 Test Environment

The video was recorded with a Sony Digital8 camcorder in standard PAL format with a image size of 720 pixels by 576 pixels. The Digital8 video standard is based on DV streams with MPEG-2 intraframe compression at a constant rate of 1:5 (e.g., [Utz 2000]).

An approximately 16 seconds long video clip was captured using a Firewire interface (IEEE.1394 standard) and Apple's iMovie software. The video clip was stored as raw DV source with the original image size of 720 pixels by 576 pixels at 25 frames per second (FPS) and 24 bit color depth. The corresponding audio track is uncompressed at 44.1 kHz, 16 bit, stereo.

Apple's Quicktime 6 software was used to convert the raw DV file to several MPEG-4 files with different settings. The following configurations were tested; an overview of the results together with some examples is presented in tables 1 and 2:

- MPEG-4 Basic, High-Res: the video with full PAL image size and frame rate is encoded using the MPEG-4 Basic compression ("ISMA Profile 0"). This codec is supported by all MPEG-4 players. The audio track is compressed with MPEG-4's AAC algorithm at a bit rate of 128 kbps (equivalent to CD quality).
- MPEG-4 Improved, High-Res: the video data is compressed with an improved compression algorithm, known as "ISMA Profile 1." The compression with this algorithm takes longer but produces better quality at the same data rate (hence producing the same file size).

The disadvantage of the improved MPEG-4 codec is that only new decoders support it. Moreover, the decoder has to be more powerful to be able to decode a file without loss.

- MPEG-4 Improved, Low-Res 1: for this test, the improved MPEG-4 compression algorithm was applied to the original DV stream with a lower image size.
- MPEG-4 Improved, Low-Res 2: in addition to the reduced image size, the frame rate is decreased to 15 frames per second, and the compression rate is increased in order to reduce the data rate.
- MPEG-4 Improved, Low-Res 3: basically, the same settings as in the previous configuration are used. However, the compression rate is further increased.

B.3 Test Results and Conclusion

The tests show that the improved compression algorithm produces better results than the basic one. However, the data rate of 264 kB/s is too small for both the basic and the improved algorithm to produce excellent image quality.

The same data rate used with a reduced image size results in an excellent picture quality (see table 2). Although a further compression is possible the picture quality deteriorates rapidly, as the last two tests show.

Therefore the configuration "MPEG-4 Improved, Low-Res 1" is recommended for use with mobile devices. It offers an acceptable data rate while the picture (and sound) quality is very good.

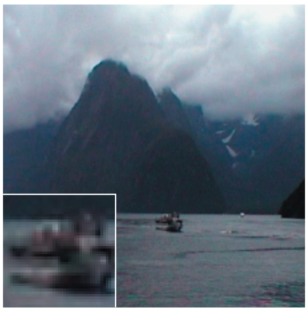
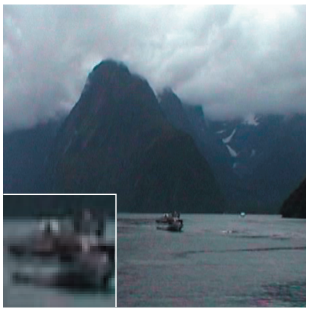
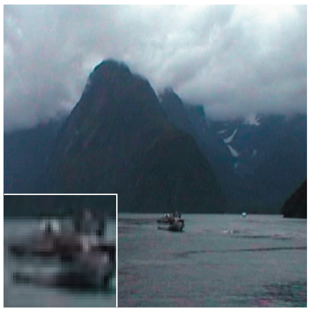
Name	DV	MPEG-4 Basic, High-Res	MPEG-4 Improved, High-Res
Codec	DV-PAL, Raw Source	MPEG-4 Basic (ISMA Profile 0)	MPEG-4 Improved (ISMA Profile 1)
Video	720 x 576, 25 FPS, 24 bit colors, MPEG-2 intraframe compression	720 x 576, 25 FPS, 24 bit colors, MPEG-4 basic compression	720 x 576, 25 FPS, 24 bit colors, MPEG-4 improved compression
Audio	44.1 kHz, 16 bit, stereo, uncompressed audio	44.1 kHz, 16 bit, stereo, AAC compressed at 128 kbps	44.1 kHz, 16 bit, stereo, AAC compressed at 128 kbps
Data Size	114.6 MB	4.3 MB	4.3 MB
Data Rate	6.8 MB/s	263.7 kB/s	263.7 kB/s
Comment	Original DV stream from a digital video camera.	Good image quality due to a medium-level compression rate.	Same size as MPEG-4 Basic but better image quality.
Example			

Table 1: Test results for MPEG-4 compressed video at a high resolution compared to the original DV stream.

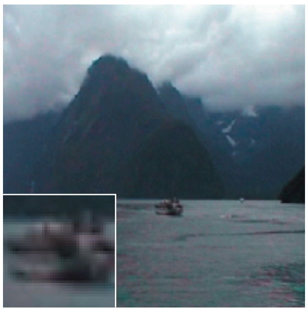
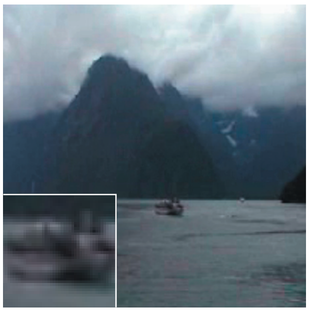
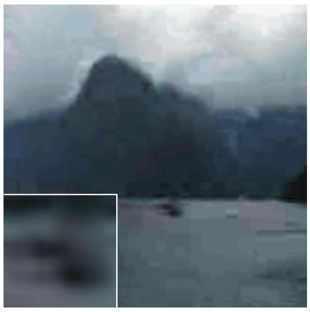
Name	MPEG-4 Improved, Low-Res 1	MPEG-4 Improved, Low-Res 2	MPEG-4 Improved, Low-Res 3
Codec	MPEG-4 Improved (ISMA Profile 1)	MPEG-4 Improved (ISMA Profile 1)	MPEG-4 Improved (ISMA Profile 1)
Video	320 x 240, 25 FPS, 24 bit colors, MPEG-4 improved compression	320 x 240, 15 FPS, 24 bit colors, MPEG-4 improved compression	320 x 240, 15 FPS, 24 bit colors, MPEG-4 improved compression
Audio	32 kHz, 16 bit, stereo, AAC compressed at 128 kbps	44.1 kHz, 16 bit, mono, AAC compressed at 96 kbps	44.1 kHz, 16 bit, mono, AAC compressed at 96 kbps
Data Size	4.1 MB	726.3 kB	308 kB
Data Rate	253.3 kB/s	43.4 kB/s	17.4 kB/s
Comment	Reduced video size. Very good image quality due to high data rate.	Same video size as before but higher compression. Lower image quality.	Very high compression, shows artifacts. Poor quality.
Example			

Table 2: Test results for MPEG-4 compressed video at higher compression rates. The results show lower image quality.

Related Documents

C

Appendix C contains several documents that are directly related to the concepts presented in this thesis: a poster, a short research report, and presentation slides.

The poster gives a brief overview of the features of the active digital video broadcasting system that is presented in Part 3 of the thesis. Although it describes the basic idea and the key features of the MPEG-4-based architecture, it does not deal thoroughly with the technical aspects.

This poster was submitted to a Poster Competition of the Department of Computer Science at the University of Auckland in December 2002.

The subsequent project report briefly summarizes the system proposed in Part 3 of the thesis. After describing the basic idea of advanced features for digital video broadcasting systems, the theoretical and technical background on which the architecture is founded is established. Finally, the MPEG-4-based approach (see section 12.3) of Vivid briefly outlined.

The project summary was handed in as application for the competition “Crazy Ideas” at Graz University of Technology in February 2003.

The slides shown in the third part of Appendix C were compiled for a one-hour lecture in which the research project “Active Digital Video Broadcasting” (previous project title “Advanced Features for Digital Video Broadcasting”) was presented. First, the family of MPEG specifications as well as the technologies and standards used in digital television environments are outlined. Subsequently, Vivid and some details of its implementation are detailed.

The lecture was given to 3rd year Masters students at the Department of Computer Science at the University of Auckland in April 2003.

This page shows a poster that presents an overview of the basic idea of the proposed system for active digital video broadcasting as described in Part 3 of the thesis.

The original size of the poster is A1 (841 mm x 594 mm). It was submitted to the 2002 Poster Competition of the Department of Computer Science at the University of Auckland. Awarded second prize.

The following two pages show a short research report (executive overview) about the MPEG-4-based architecture of Vivid.

It was submitted to "Crazy Ideas", an innnovative ideas competition carried out by Graz University of Technology and sponsored by the Bank Austria-Creditanstalt in March 2003. Awarded first prize.

Josef Kolbitsch
jkol005@cs.auckland.ac.nz

Advanced Features for Digital Video Broadcasting

Dr. Jennifer A. Lennon
Department of Computer Science
University of Auckland

Dr. Hermann Maurer
Institute for Information Processing and Computer-Supported New Media (IICM)
Graz, University of Technology

In which movie have I seen this actress before?

Do questions such as *"In which movie have I seen this actress before?"* or *"What is the title of the song in the background?"* sound familiar to you?

Today's digital video systems – DVDs, digital television, computer-based video players, etc. – are not capable of answering these questions. Emerging multimedia standards are focusing on the description of digital content and digital rights management (see the box "The Family of MPEG Standards").

This project employs those novel techniques and proposes an advanced media application that is able to answer many of the users' questions. In doing so, it provides enhanced interactivity to the user.

Objects

Whenever an object is selected, basic information about it is displayed. This information contains elements such as the content type (video in this example), the encoding, and the data size.

This is essentially the information that most of the current media players can provide.

Full advantage of the advanced player application can be taken, if MPEG-4 objects are used.

Object Descriptors

The different objects in a scene are described using MPEG-7, which enables content-based retrieval. The user could easily run queries such as *"find all scenes in which this actress occurs throughout the movie"*. The user could also tell the system to query a dedicated movie database, IMDB for instance, to find out more about the actor.

Digital Rights

Digital rights are necessary to protect the content and the intellectual property. They are necessary for the user and the system to understand what can be done with the content and who owns it – How often may it be viewed? May it be printed or copied, may it be saved to an external storage device?

The Family of MPEG Standards

MPEG is well known for its coding standards for digital media including MPEG-1 (VideoCD, MP3) and MPEG-2 (DVD). MPEG's most recent standard for representing digital media is MPEG-4, in which content is described using *objects* that can be of virtually any shape and any kind of media.

However, MPEG does not only specify standards for *representing* content but also for *describing* the content. The new MPEG-7 standard is a "Multimedia Content Description Interface", whose purpose it is to characterize features on different levels of the content: low-level features include color histograms (for pictures), pitch (for sound), and motion vectors (for moving objects). On higher levels structural information, relations among objects, semantic features, and bibliographic data are expressed. The most important application of MPEG-7 is expected to be content-based retrieval (CBR).

MPEG-21 is a framework for enabling Intellectual Property Management and Protection (IPMP). It is still under development and has not been ratified as a standard. The central aspects will be the identification and definition of digital objects and their corresponding digital rights, the different roles of actors and their permissions, as well as the effective protection of content.

Brief System Overview

The MPEG-4 media objects are stored together with the corresponding descriptors and digital rights in a database. In organizations such as TV stations, the metadata is often readily available, frequently stemming from the production process.

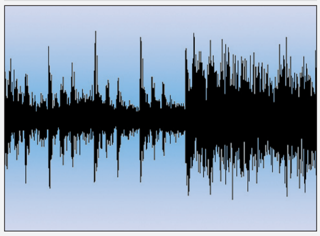
For streaming video broadcasts over the internet, the transmission is straight forward: MPEG-4 objects are multiplexed with MPEG-7 and -21 data. For devices not equipped with MPEG-4 (e.g., set-top boxes), an "MPEG-2 wrapper" is utilized. An MPEG-2 stream that is supplemented with special packets carrying the metadata is transmitted. For these devices, not all features might be available.

The complete scene in the first picture consists of three objects: the background (a still image), the actress (a moving object), and the sound. These objects are in a spatio-temporal relationship to each other, which is described using MPEG-4.

Since the objects are independent entities, descriptors (MPEG-7) and digital rights (MPEG-21) can be attached individually. The description of the actress in the picture below could contain her name and a hyperlink to her website.

When her name is available it can be used to search the internet or a movie database. Hence, the user can easily find an answer to his question *"Where have I seen her before?"* The user could also request a summary of all scenes of the movie in which she occurs.

You like the music in the background of the movie? The MPEG-21 coded digital rights tell you that you bought it together with the movie. You can download it to your PC and listen to it. However, a copy-protection mechanism prevents you from giving it to someone else.



in which movie *In which movie* in which movie *have I seen* *this actress* this actress *before?*

Advanced Features for Digital Video Broadcasting

Josef Kolbitsch

Introduction

Do questions such as *"In which movie have I seen this actress before?"* or *"What is the title of the song in the background?"* sound familiar to you? Today's digital video systems such as DVDs, digital television (DTV), and computer-based video players are not capable of answering these questions.

This project utilizes new techniques for describing content and proposes a system for digital video broadcasting environments that makes it possible to provide explanations for many of the questions above. In doing so, it offers enhanced interactivity to the consumer.

The system can be implemented for use with DTV, conventional computer networks, and ubiquitous computing devices such as mobile phones or handheld devices.

Scenarios and Examples

The proposed system can be employed with a wide variety of different content types including movies, news and documentaries, sports, and children's television. The following few paragraphs give an overview of usage scenarios and provide a few examples.

Movies

A person is watching a movie on TV and wonders where he has seen this particular actress before. The viewer uses a pointer device built into the remote control and selects the actress on the screen. A window pops up and presents information about the actress: her name and role in the movie, her real name, other data, and a reference to a specialized movie database such as the Internet Movie Database (IMDb). A link to her personal website might also be available.

In a similar way the viewer can find out which other movies the director has directed, who has written the book or script, whether this film has been awarded, etc.

Another example focuses on the soundtrack of a movie: the user likes a particular song and would like to have it. He requests information on this scene, a window on the screen pops up and lists all noteworthy elements of the scene. It includes all actors and audio clips. Now the user knows the name of the song and the artist and can also choose to download it instantly to the set-top box, computer, etc. This scenario is illustrated in figure 1.

News and Documentaries

News and documentaries can sometimes be quite difficult to follow if one is not familiar with the context, domain specific facts, or the historical background. In this case references to related "articles" or background information might be very useful. Basically, this is an analogy to many news services found on the WWW.

In a news broadcast the proposed system can be used to offer more specific details for interested viewers. A news item about the Columbia space shuttle disaster, for instance, can be supplemented with a graph showing the flight path, a chronology of the accident, related news stories, etc. A hyperlink to the Nasa website complements the report.

Documentaries can use similar features: imagine a school class watching a film about

This project is part of a Master Thesis in Telematics at the Institute for Information Processing and Computer-Supported New Media (IICM) at Graz, University of Technology, Austria, and at the Hypermedia Unit (HMU) of the Department of Computer Science at the University of Auckland, New Zealand.

World War II. The teacher wants to explain some events in detail and therefore he requests additional information from the system. A variety of audio documents, other video clips, maps, text documents, etc. is readily available. The teacher selects a historical map, and has it displayed on the screen. Alternatively he could download it and print it out.

Economic Aspects

The efficiency of the system is impressively presented in the example with the movie soundtrack. Traditionally the user would have to learn the name of the song and on which CD to find it. This can sometimes be quite demanding. Then he has to find a record store that has the CD in stock, and finally ends up buying the complete album although he only wanted to have one song. This is far too complicated and most people are either deterred by the effort or simply forget about the song.

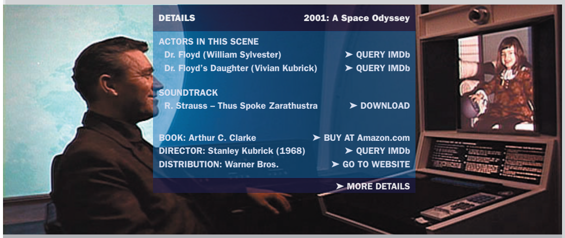


Figure 1: A scene from *"2001: A Space Odyssey"* with some details.

However, with the proposed system the user can instantly download the song without any inconvenience. Moreover valuable additional information is available. Hence, the system provides immediate benefits for consumers: content that might otherwise be hard to find is delivered in an easy-to-use and intuitive way.

Cooperations with companies such Amazon.com can result in profits for both the TV network provider and the content distributor or service provider.

Technical Background

The Family of MPEG Standards

MPEG is well known for its coding standards for digital media including MPEG-1 (VideoCD, MP3) and MPEG-2 (DVD, DTV). MPEG's most recent standard for representing digital media is MPEG-4, in which content is described using *objects* that can be of virtually any shape and any kind of media.

However, MPEG does not only specify standards for *representing* content but also for *describing* the content. The new MPEG-7 standard is a "Multimedia Content Description Interface", whose purpose is to characterize features on different levels of the content: low-level features include colour histograms (for pictures), pitch (for sound), and motion vectors (for moving objects). On higher levels structural information, relations among objects, semantic features, and bibliographic data are expressed. The most important application of MPEG-7 is expected to be content-based retrieval (CBR).

MPEG-21 is a framework for enabling Intellectual Property Management and Protection (IPMP). It is still under development and has not been ratified as a standard. The central aspects will be the identification and definition of digital objects and their corresponding digital rights, the different roles of actors and their permissions, as well as the effective protection of content.

Digital Video Broadcasting

The most widely accepted standard for digital television systems is the Digital Video Broadcasting specification (DVB) defined by the European Telecommunication Standards Institute (ETSI). It is based on MPEG-2 and deals with aspects such as transport and return channels, data transmission, and terminal devices.

The Multimedia Home Platform (MHP) is a framework that enables applications to run in a standardized environment on top of DVB. It is an interface between (interactive) applications and the terminal hardware on which they are executed. Thus, application development is greatly facilitated.

Popular examples for MHP applications include electronic program guides, information services such as stock and news tickers, games, and e-commerce projects.

Brief System Overview

The system associates a set of metadata and digital rights with every media object in the system. Content is preferably encoded with MPEG-4, the corresponding content descriptors are specified with MPEG-7, and digital rights are defined using MPEG-21.

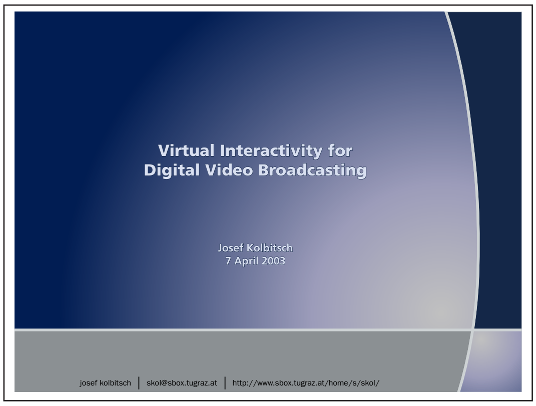
In computing environments the transmission process is straightforward: an MPEG-4 stream is multiplexed with MPEG-7 and -21 data and transmitted via a traditional network connection.

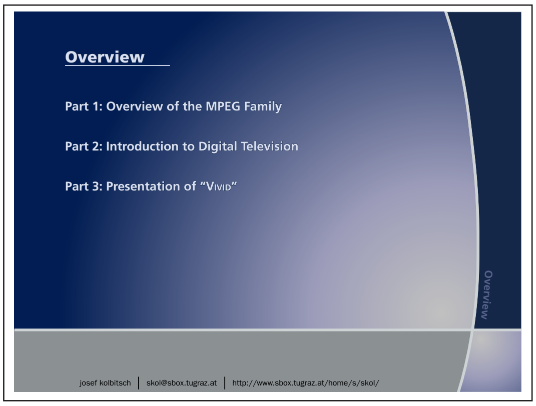
For MPEG-2-based DTV an "MPEG-2 wrapper" that transcodes MPEG-4 objects to MPEG-2 is utilized. The MPEG-2 stream is supplemented with special packets carrying metadata and digital rights and is delivered using DVB.

Conclusion

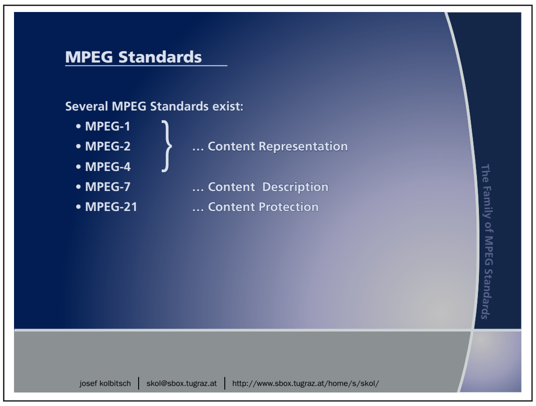
The project applies most recent technologies such as MPEG-7, MPEG-21, and MHP and describes a system that aims at DTV as well as computer-based and mobile devices. The advanced features for digital video broadcasting bring interactivity to the user and result in immediate benefits for the consumer.

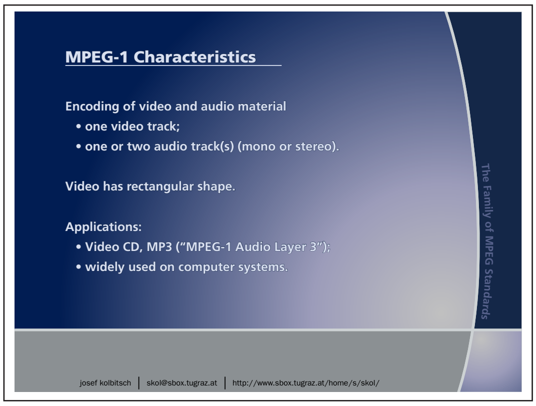
The following eight pages contain the sildes for a presentation of VIVID, the system proposed in Part 3 of the thesis. The presentation was given in April 2003.

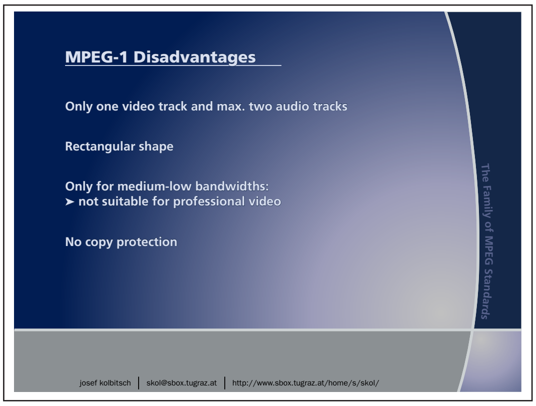

Virtual Interactivity for
Digital Video Broadcasting
Josef Kolbitsch
7 April 2003
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Overview
Part 1: Overview of the MPEG Family
Part 2: Introduction to Digital Television
Part 3: Presentation of "Vivid"
Overview
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


The Family of MPEG Standards
Part 1
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


MPEG Standards
Several MPEG Standards exist:
• MPEG-1
• MPEG-2
• MPEG-4
… Content Representation
• MPEG-7 … Content Description
• MPEG-21 … Content Protection
The Family of MPEG Standards
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


MPEG-1 Characteristics
Encoding of video and audio material
• one video track;
• one or two audio track(s) (mono or stereo).
Video has rectangular shape.
Applications:
• Video CD, MP3 ("MPEG-1 Audio Layer 3");
• widely used on computer systems.
The Family of MPEG Standards
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


MPEG-1 Disadvantages
Only one video track and max. two audio tracks
Rectangular shape
Only for medium-low bandwidths:
➤ not suitable for professional video
No copy protection
The Family of MPEG Standards
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-2 Characteristics

Basically the same approach as MPEG-1, but:
- several video tracks (not only one);
- several audio tracks (not only two),
  - ➤ surround sound "5.1 Audio";
- supports high bandwidths
  - ➤ suitable for professional video.

Applications:
- DVD;
- Digital Television.

MPEG-2 Advantages

Solves some problems of MPEG-1

Designed to provide numerous "profiles":
- one profile for professional film production, e.g., Star Wars Episode 1;
- one profile for digital television;
- one profile for DVDs;
- one profile for home users.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-2 Disadvantages

MPEG-2 is designed for high bandwidth environments:
- digital telvision and DVDs.

Does not allow for low-bandwidth networks:
- Internet and WWW,
- mobile devices, etc.

Still no copy protection

Still rectangular shape for video content

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-4

MPEG-1 → MPEG-2 → MPEG-4?

What happened to MPEG-3? Is "MP3" MPEG-3?
- MP3 is *not* MPEG-3.
  - ➤ MP3 stands for "MPEG-1 Audio Layer 3."
  - ➤ Hence, it is part of MPEG-1.
- MPEG-3 was meant to be the standard for HDTV

MPEG-2 is so flexible that MPEG-3 became obsolete.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-4 Characteristics

Completely new approach to content encoding

Can do everything MPEG-1 or MPEG-2 can do

Additionally:
- Much higher compression rates;
- Video can be of virtually any shape;
- Also suitable for low-bandwidth networks;
- Design has mobile devices and networked computers in mind;
- Supports Intellectual Property Management and Protection.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

The MPEG-4 Approach

Every video clip consists of scenes

Every scene contains a number of objects:
- still images;
- short video clips (not necessarily rectangular!);
- short audio clips, etc.

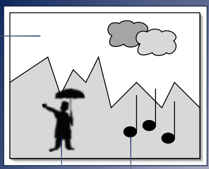
Scene description specifies which object is on which position and when it occurs.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Example for an MPEG-4 Scene

Background:
still image



Person:
moving object
(i.e., video clip)

Sound track:
audio clip

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-4 Applications

Might become standard for multimedia on the Internet.

Used on computers:
- “Real” MPEG-4: e.g., Apple’s Quicktime software.
- MPEG-4 compatible: DivX, etc.

Used for mobile devices:
- Handheld computers:
  Sanyo produces hardware,
  MPEG-4 player for Linux handelds;
- Mobile phones: e.g., Siemens.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-7

MPEG-1, MPEG-2, and MPEG-4 for content *representation:*
How to compress, encode, store, transmit content.

MPEG-7 for content *description* – metadata.

Examples:
- Bibliographic, high-level: What is a movie about?
- Structural: object A and object B are in the same scene.
- Low-level: color histogram, pitch.

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

MPEG-21

MPEG-21 for Intellectual Property Management and Protection

Objectives:
- Identify objects,
  e.g., a book can be uniquely identified with its ISBN;
- Describe objects,
  e.g., author of the book, number of pages, etc.;
- Specify intellectual property information:
  who owns the content, who distributes it, copyright, ...;
- Define digital rights:
  may content be copied? how many times viewed? ...

The Family of MPEG Standards

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Digital Television (DTV)

Part 2

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Why Digital Television?

More TV channels
Better picture quality
Better sound quality (sourround sound)

Can offer additional services such as:
- Internet access;
- e-Mail;
- shopping and, generally, e-Commerce;
- games; etc.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Digital vs. Analog Standards

Analog TV standards:
- PAL: Europe, Australia, NZ, large parts of Asia, Russia, etc.;
- NTSC: USA, Canada.

Digital TV standards:
- ATSC: USA, Canada, South Korea;
- ISDB (very similar to DVB): Japan;
- DVB: rest of the world.

Both ATSC, ISDB, and DVB are based on MPEG-2.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

DVB

DVB stands for "Digital Video Broadcasting" Specification

DVB standard describes how MPEG-2 is to be used:
- Which profile for production (high quality);
- which profile for broadcasting (medium quality);
- how is MPEG-2 data transmitted;
- what does a digital TV set have to do with the MPEG-2 stream;
- other details of the TV production and broadcasting process.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

DVB: Transmission

The DVB Standard defines transmission of:
- video streams;
- audio streams; and
- digital data.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

DVB: Digital Data

Digital data can contain:
- electronic program guides (EPGs);
- teletext;
- subtitles;
- other digital data; and
- application programs – software!

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

DVB: MHP

The DVB Specification describes a standard for developing software on top of DVB.

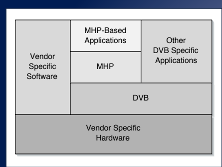
The Multimedia Home Platform – MHP.

MHP offers a Java Virtual Machine:
- can run Java applications;
- offers APIs for access to screen and video, audio, and data streams.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Summary/Overview of Standards



Applications rely on MHP

MHP relies on DVB

DVB relies on MPEG-2

Reason for complexity: interoperability!
You want to buy a TV set from *any* vendor, plug it in, and use it.

Digital Television

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

VIVID

Part 3

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

What is VIVID?

VIVID is a system for **V**irtual **I**nteractivity in **Vid**eo Broadcasting.

VIVID brings enhanced interactivity to consumers.

VIVID introduces features of Knowledge Management in classical video broadcasting environments.

Vivid

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

What does that mean? – An Example

Imagine you are watching TV, a movie or TV series.

You see an actress.

You know you have seen her before … but who is she?
And in which movie have you seen her before?

How would you usually find answers to these questions?

Vivid

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Another Example

You are watching a movie.
You hear a song in the background.
You like it and would like to have it.

What is the name of the song?
The name of the artist?
The name of the album? And where to get it …

Is there an easier way?

Vivid

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Aim of VIVID

VIVID makes video broadcasting interactive by permitting users to find answers to their questions.

And it can do more:

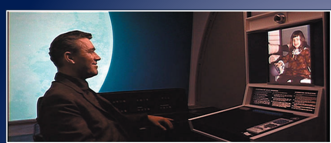
- allows downloading of content such as the sound track, video clips, etc.;
- provides links to specialized databases;
- provides links to related “articles”, etc.

VIVID is not only for DTV but for all kinds of digital video broadcasting!

Vivid

josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

Example: Digital Television

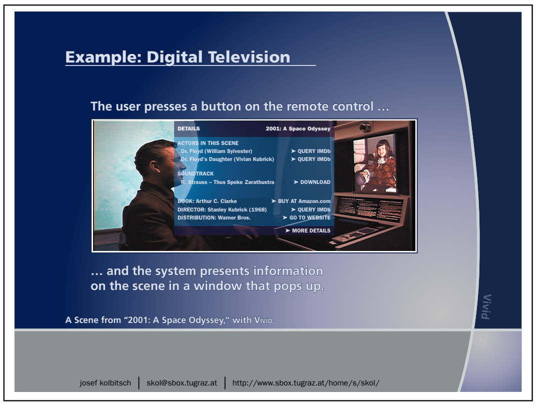


The user would like to know who the actor is.

A Scene from “2001: A Space Odyssey,” without VIVID.

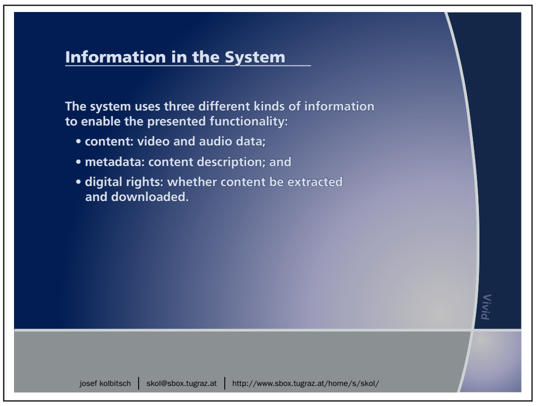
Vivid

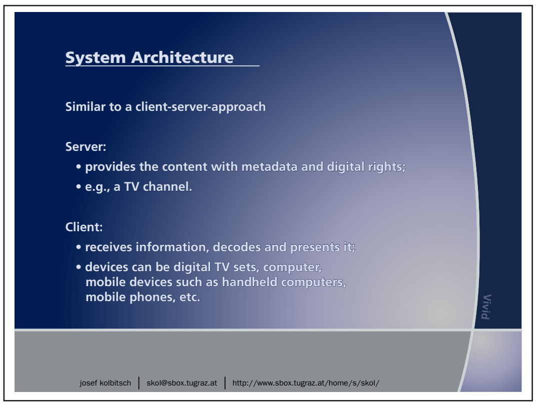
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

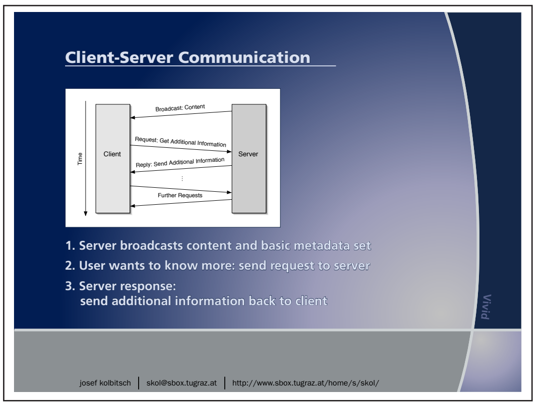

Example: Digital Television
The user presses a button on the remote control …
… and the system presents information
on the scene in a window that pops up.
A Scene from "2001: A Space Odyssey," with Vivid.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

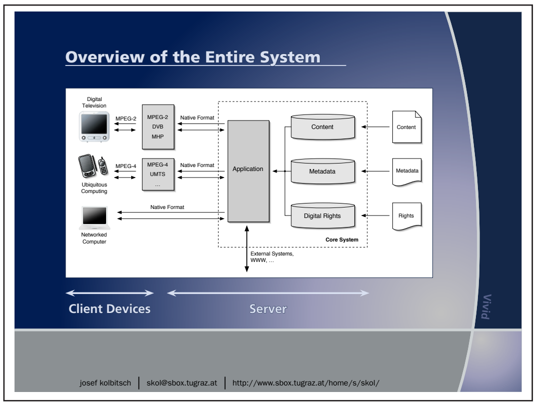

Example: Computer-Based Video
Advanced Player Prototype
Information
Store Object
The user watches a video
broadcast on the computer.
Seemingly, a conventional
media player.
But two additional buttons:
"Information" and
"Store Object"
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

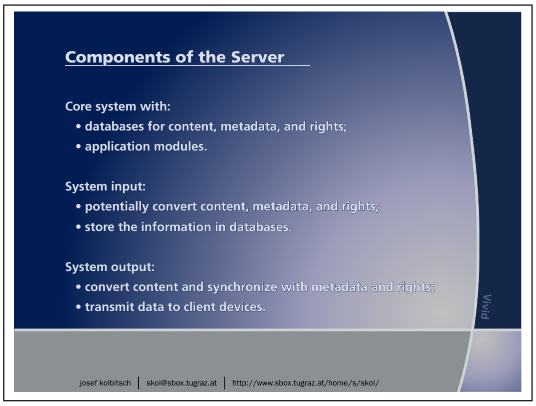

Example: Computer-Based Video
Advanced Player Prototype
Selected Object
Object Description
Digital Rights
Information
Store Object
User clicks on
an object and
presses "Infor-
mation".
Vivid displays
a basic set of
metadata.
"Store" is deac-
tivated because
content is copy-
protected.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

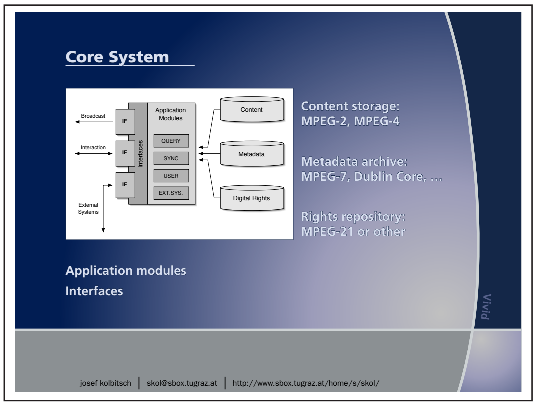

Information in the System
The system uses three different kinds of information
to enable the presented functionality:
• content: video and audio data;
• metadata: content description; and
• digital rights: whether content be extracted
and downloaded.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

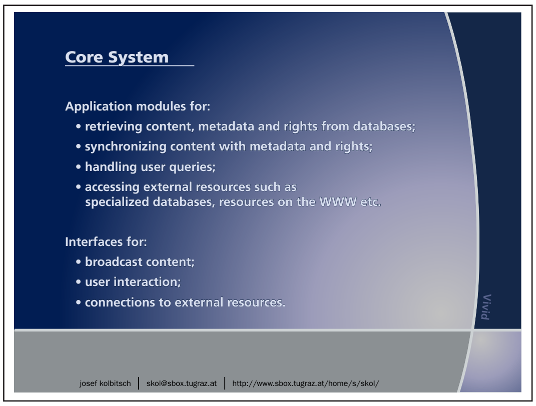

System Architecture
Similar to a client-server-approach
Server:
• provides the content with metadata and digital rights;
• e.g., a TV channel.
Client:
• receives information, decodes and presents it;
• devices can be digital TV sets, computer,
mobile devices such as handheld computers,
mobile phones, etc.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

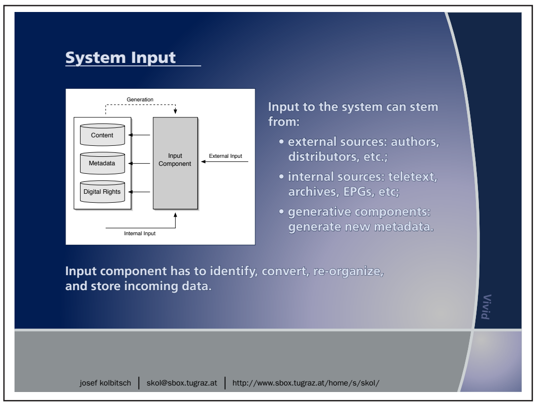

Client-Server Communication
Time
Client
Server
Broadcast: Content
Request: Get Additional Information
Reply: Send Additional Information
Further Requests
1. Server broadcasts content and basic metadata set
2. User wants to know more: send request to server
3. Server response:
send additional information back to client
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

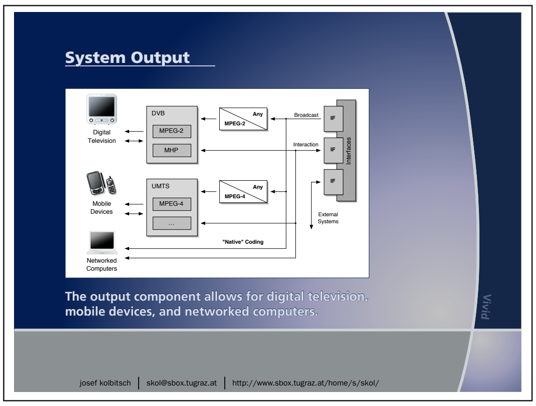

Overview of the Entire System
Client Devices
Server
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

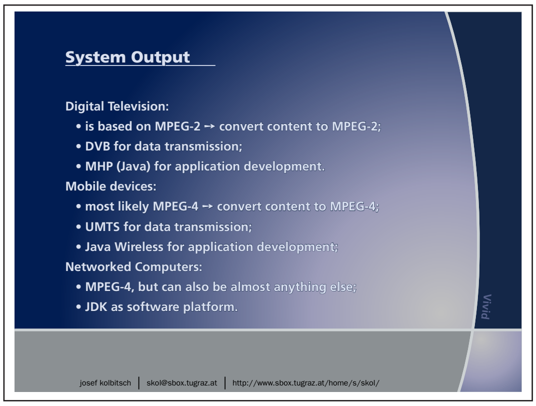

Components of the Server
Core system with:
• databases for content, metadata, and rights;
• application modules.
System input:
• potentially convert content, metadata, and rights;
• store the information in databases.
System output:
• convert content and synchronize with metadata and rights;
• transmit data to client devices.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

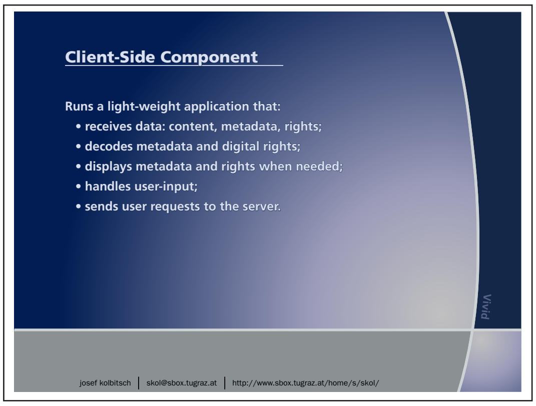

Core System
Content storage:
MPEG-2, MPEG-4
Metadata archive:
MPEG-7, Dublin Core, ...
Rights repository:
MPEG-21 or other
Application modules
Interfaces
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

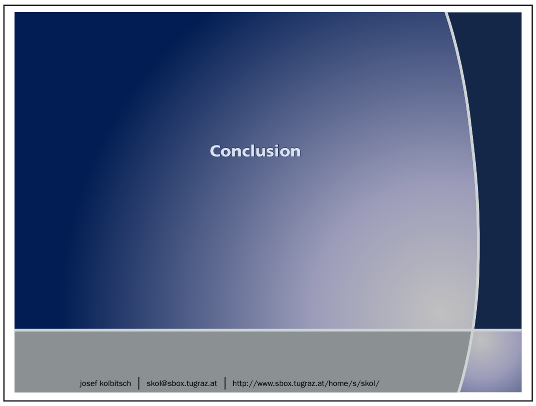

Core System
Application modules for:
• retrieving content, metadata and rights from databases;
• synchronizing content with metadata and rights;
• handling user queries;
• accessing external resources such as specialized databases, resources on the WWW etc.
Interfaces for:
• broadcast content;
• user interaction;
• connections to external resources.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

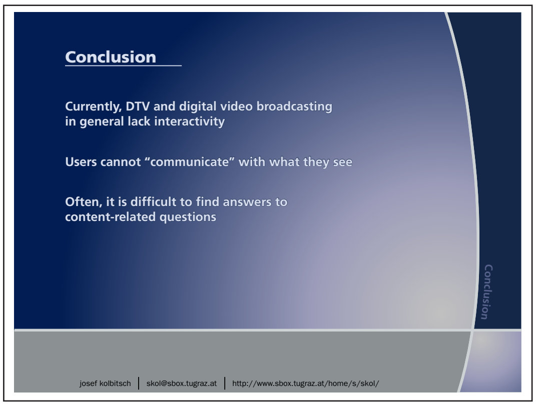

System Input
Input to the system can stem from:
• external sources: authors, distributors, etc.;
• internal sources: teletext, archives, EPGs, etc;
• generative components: generate new metadata.
Input component has to identify, convert, re-organize, and store incoming data.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

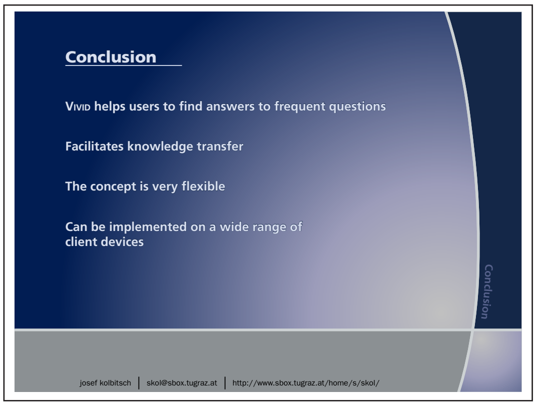

System Output
The output component allows for digital television, mobile devices, and networked computers.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


System Output
Digital Television:
• is based on MPEG-2 → convert content to MPEG-2;
• DVB for data transmission;
• MHP (Java) for application development.
Mobile devices:
• most likely MPEG-4 → convert content to MPEG-4;
• UMTS for data transmission;
• Java Wireless for application development;
Networked Computers:
• MPEG-4, but can also be almost anything else;
• JDK as software platform.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Client-Side Component
Runs a light-weight application that:
• receives data: content, metadata, rights;
• decodes metadata and digital rights;
• displays metadata and rights when needed;
• handles user-input;
• sends user requests to the server.
Vivid
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Conclusion
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Conclusion
Currently, DTV and digital video broadcasting
in general lack interactivity
Users cannot "communicate" with what they see
Often, it is difficult to find answers to
content-related questions
Conclusion
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Conclusion
Vivid helps users to find answers to frequent questions
Facilitates knowledge transfer
The concept is very flexible
Can be implemented on a wide range of
client devices
Conclusion
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/


Thank you for your attention.
All content and layout by Josef Kolbitsch.
7 April 2003
josef kolbitsch | skol@sbox.tugraz.at | http://www.sbox.tugraz.at/home/s/skol/

References

[AAAI 2002] American Association for Artificial Intelligence, 2002, *Case-Based Reasoning,* http://www.aaai.org/AITopics/html/casebased.html, Visited October 14th, 2002.

[Aamodt and Plaza 1994] Aamodt, A., and Plaza, E., 1994, *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches,* Artificial Intelligence Communications (AICom), Volume 7, Number 1, pp. 39-59. See also http://www.iiia.csic.es/People/enric/AICom.html.

[Abecker et al. 1998] Abecker, A., Bernardi, A., and Sintek, M., 1998, *Proactive Knowledge Delivery for Enterprise Knowledge Management.* In: Stuart Barnes (ed.), *Knowledge Management Systems: Theory and Practice,* International Thomson Business Press, 2001.
See also http://www.dfki.uni-kl.de/~aabecker/Postscript/ECIS99/final-ecis99-word8.htm.

[AGMA 2003] Justus-Liebig-Universität Giessen, Fachbereich Medizin, 2003, *Abfrage Medizinischer Lernprogramme,* http://www.agma.med.uni-giessen.de/cfagma/lern/lernresults.dbm?text=&max=999, Visited April 12th, 2003.

[Agnew and Kniesner 2001] Agnew, G., and Kniesner, D. (eds.), 2001, *ViDe User's Guide: Dublin Core Application Profile for Digital Video,* http://www.vide.net/workgroups/videoaccess/resources/vide_dc_userguide_20010909.pdf, Visited November 17th, 2002.

[Aha 2002a] Aha, D. W., 2002, *NCARAI Interactive Decision Aids Group,* http://www.aic.nrl.navy.mil/~aha/cbr/, Visited October 22nd, 2002.

[Aha 2002b] Aha, D. W., 2002, *Intelligent Decision Aids,* Presentation.
See also http://www.aic.nrl.navy.mil/~aha/ida/Group-Info/Presentation-2-02.ppt.

[Ahamad 2002] Ahamad, I., 2002, *Multimedia Technology Research Center,* http://www.mtrec.ust.hk/, Visited February 5th, 2003.

[AI-CBR 2002] Watson, I., et al., 2002, *Case-Based Reasoning: The AI-CBR Homepage,* http://ai-cbr.cs.auckland.ac.nz/, Visited October 14th, 2002.

[Aitken 2001] Aitken, S., 2001, *Case-Based Reasoning at AIAI,* http://www.aiai.ed.ac.uk/project/cbr/, Visited October 23rd, 2002.

[Alavi and Leidner 1999] Alavi, M., and Leidner, D. E., 1999, *Knowledge Management Systems: Issues, Challenges, and Benefits,* Communications of the Association for Information Systems (AIS), Volume 1, February 1999, Article 7.

[Allen 2002] Allen, J., *Knowledge Management and Speech Recognition,* Computer, Volume 35, Issue 3, March 2002, pp. 60-61.

[Althoff et al. 1998] Althoff, K.-D., Nick, M., and Tautz, C., 1998, *Concepts for Reuse in the Experience Factory and their Implementation for CBR-System Development,* Proceedings of the 1998 Treffen der GI-Fachgruppe 1.1.3 "Maschinelles Lernen". See also http://ki.cs.tu-berlin.de/~fgml98/postscript/althoff.ps.

[Apple 2002] Apple Computers, Inc., 2002, *Apple,* http://www.apple.com/, Visited November 10th, 2002.

[Arentz 2001] Arentz, W. A., 2001, *Beat Extraction from Digital Music,* http://ilab0.ux.his.no/norsig/presentation/44.Beat_Extract_10102001202111.ppt, Visited April 10th, 2003.

[Atnafu et al. 2002] Atnafu, S., Chbeir, R., Brunie, L., 2002, *Efficient Content-Based and Metadata Retrieval in Image Database,* Journal of Universal Computer Science, Volume 8, Issue 6, June 2002, pp. 613-622.
See also http://www.jucs.org/jucs_8_6/efficient_content_based_and/.

[Avaro and Salembier 2001] Avaro, O., and Salembier, P., 2001, *MPEG-7 Systems: Overview,* IEEE Transactions on Circuits and Systems for Video Technology, Volume 11, Number 6, June 2001, pp. 760-764. See also http://gps-tsc.upc.es/imatge/pub/ps/IEEE_CSVT2001_Avaro_Salembier.pdf.

[Babilon 1998] Babilon, M., 1998, *The Evolution of Knowledge Management within NCR Corporation,* Proceedings of the Sixteenth Annual International Conference on Computer Documentation, September 1998.

[Bagliani 2002a] Bagliani, A., Sony Electronics, Inc., 2002, *The MPEG IMX™ Format and the Digital Production of Motion Pictures.* See also http://www.sonympeg-imx.com/resources/MPEGIMX_wpaper.pdf.

[Bagliani 2002b] Bagliani, A., Sony Electronics, Inc., 2002, *MPEG IMX™ Recording. The Digital Heir to the Analog Betacam SP® Format.* See also http://www.sonympeg-imx.com/resources/MPEGHeir.pdf.

[Baker 2002] Baker, K., 2002, *Where Will Knowledge Management Take Us?* In: Dieng-Kuntz, R., and Matta, N. (eds.), *Knowledge Management and Organizational Memories,* pp. 3-16, Kluwer Academic Publishers, Boston 2002.

[Balabanian et al. 1996] Balabanian, V., Casey, L., Greene, N., and Adams, C., 1996, *An Introduction to Digital Storage Media – Command and Control (DSM-CC),* IEEE Communications Magazine, November 1996. See also http://mpeg.telecomitalialab.com/documents/dsmcc/dsmcc.htm.

[BBCi 2003] BBC International, 2003, http://newsimg.bbc.co.uk/media/images/38794000/jpg/_38794123_silhouette-afp-300.jpg, Visited February 8th, 2003.

[Bell 2003] Bell, G. S., 2003, *The Microsoft Agent Ring,* http://www.msagentring.org/, Visited February 9th, 2003.

[Bellinger et al. 2000] Bellinger, G., Castro, D., Mills, A., 2000, *Data, Information, Knowledge, and Wisdom,* http://www.outsights.com/systems/dikw/dikw.htm, Visited October 7th, 2002.

[Benitez and Chang 2002a] Benitez, A. B., Chang, S.-F., 2002, *Semantic Knowledge Construction from Annotated Image Collections,* Proceedings of the 2002 International Conference on Multimedia & Expo (ICME-2002), Lausanne, Switzerland, 2002. See also http://www.ctr.columbia.edu/papers_advent/02/icme02_2_ana.pdf.

[Benitez and Chang 2002b] Benitez, A. B., Chang, S.-F., 2002, *Perceptual Knowledge Construction from Annotated Image Collections,* Proceedings of the 2002 International Conference on Multimedia & Expo (ICME-2002), Lausanne, Switzerland, 2002. See also http://www.ctr.columbia.edu/papers_advent/02/icme02_1_ana.pdf.

[Benitez and Chang 2002c] Benitez, A. B., Chang, S.-F., 2002, *Multimedia Knowledge Integration, Summarization, and Evaluation,* Proceedings on the 2002 International Workshop on Multimedia Discovery & Data Mining in Conjunction with the International Conference on Knowledge Discovery & Data Mining (MDM/KDD-2002), Edmonton, Alberta, Canada, 2002. See also ftp://ftp.ee.columbia.edu/CTR-Research/advent/public/papers/02/mdm-kdd02_ana.pdf.

[Benitez and Smith 2001] Benitez, A. B., Smith, J. R., 2001, *New Frontiers for Intelligent Content-Based Retrieval,* Proceedings of the SPIE 2001 Conferene on Storage and Retrieval for Media Databases (IS&T/SPIE-2001), Vol. 4315, San Jose, CA, 2001. See also http://www.ctr.columbia.edu/papers_advent/01/SPIEjan01_ana.pdf.

[Benitez et al. 2000] Benitez, A. B., Smith, J. R., Chang, S.-F., 2000, *MediaNet: A Multimedia Information Network for Knowledge Representation,* Proceedings of the SPIE 2000 Conferene on Internet Multimedia Systems (IS&T/SPIE-2000), Vol. 4210, Boston, MA, 2000. See also http://www.ctr.columbia.edu/papers_advent/01/SPIEnov00_ana.pdf.

[Benitez et al. 2001a] Benitez, A. B., Chang, S.-F., Smith, J. R., 1998, *IMKA: A Multimedia Organization System Combining Perceptual and Semantic Knowledge,* Proceedings of the 9th ACM International Conference on Multimedia (ACM MM-2001), Canada, Ottawa, 2001. See also http://www.ctr.columbia.edu/papers_advent/01/ACMMMoct01_ana.pdf.

[Benitez et al. 2001b] Benitez, A. B., Zhong, D., et al., 2001, *MPEG-7 MDS Content Description Tools and Applications,* Proceedings of the International Conference on computer Analysis of images and Patterns (CAIP-2001), Warsaw, Poland, 2001. See also http://www.ctr.columbia.edu/papers_advent/01/CAIPsep01_ana.pdf.

[Bergin 2000] Bergin, H., WHD Public Relations, 2000, *DVB Provides the Ultimate Tool for Interactivity with the Launch of DVB-MHP at IBC,* Press Release, DVB Project Office, September 8th, 2002.

[Bionic 2003] Bionic Buffalo Corporation, 2003, *Tatanka™ DSM-CC Components and Applications,* http://www.tatanka.com/prod/info/dsmcc.html, Visited January 25th, 2003.

[Borghoff and Pareschi 1997] Borghoff, U. M., Pareschi, R., 1997, *Information Technology for Knowledge Management,* Journal of Universal Computer Science Volume 3, Issue 8, pp. 835-842. See also http://www.jucs.org/jucs_3_8/information_technology_for_knowledge/.

[Boyack et al. 2001] Boyack, K. W., Wylie, B. N., Davidson, G. S., 2001, *Information Visualization, Human-Computer Interaction, and Cognitive Psychology: Domain Visualizations,* Publication of the Sandia National Laboratories.

[Buchanan and Shortliffe 1984] Buchanan, B. G., and Shortliffe, E. H., 1984, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project,* Addison-Wesley, Reading, MA, 1984.

[Burkhard 1998] Burkhard, H.-D., 1998, *Extending Some Concepts of CBR – Foundations of Case Retrieval Nets.* In: Lenz, M., et al. (eds.), *Case-Based Reasoning Technology. From Foundations to Applications,* pp. 17-50, Springer-Verlag Berlin, 1998.

[Burow et al. 1998] Burow, R., Pogrzeba, P., Christ, P., Deutsche Telekom Berkom, 1998, *Mobile Reception of DVB-T,* FKT 11, Hüthig Verlag, Germany, 1998.

[CA 2001] Consumers' Association, 2001, *Turn on, Tune in, Switched off. Consumer Attitudes to Digital TV,* March 2001. See also http://www.dvb.org/dvb_technology/whitepaper-pdf-docs/Consumers%20Association%20Rvue%20of%20Digital%20TV.pdf.

[Casey 2001] Casey, M., 2001, *General Sound Recognition and Similarity Tools,* http://www.tnt.uni-hannover.de/project/mpeg/audio/general/aes110_5_SoundRecog.pdf, Visited April 25th, 2003.

[Chang and Znati 2001] Chang, S.-K., and Znati, T., 2001, *Adlet: An Active Document Abstraction for Multimedia Information Fusion,* IEEE Transactions on Knowledge and Data Engineering, Volume 13, Issue 1, January/February 2001, pp. 112-123.

[Chang et al. 1997] Chang, S.-F., Chen W., et al., 1997, *VideoQ: An Automated Content Based Video Search System Using Visual Cues,* ACM Multimedia 97. See also http://www.ee.columbia.edu/~dzhong/Papers/acm97.pdf.gz.

[Chen 2001] Chen, C., 2001, *Visualization of Knowledge Structures.*

[Chen et al. 1999] Chen, A. L. P., Chin-Chin, L., and Kuo, T. C. T., 1999, *Content-Based Video Data Retrieval,* Proceedings of the National Science Council, Volume 23, Number 4, 1999, pp. 449-465. See also http://nr.stic.gov.tw/ejournal/ProceedingA/v23n4/449-465.pdf.

[Cherry 2001] Cherry, S. M., *Making Music Pay,* IEEE Spectrum, Volume 38, Issue 10, October 2001, pp. 41-46.

[Cherry 2002] Cherry, S. M., *Getting Copyright Right,* IEEE Spectrum, Volume 39, Issue 2, February 2002, pp. 47-51.

[Cheung and Curreem 2002] Cheung, S. C., and Curreem, H., 2002, *Rights Protection for Digital Contents Redistribution Over the Internet,* Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02), 2002.

[Chiariglione 2002] Chiariglione, L., Telecom Italia Lab, 2002, *Introduction to MPEG-2: Multimedia Content Description Interface.* See also http://www.wileyeurope.com/cda/cover/0,,0471486787%7Cexcerpt,00.pdf.

[Choo et al. 1998] Choo, C. W., Detlor, B., Turnbull, D., 1998, *The Data-Information-Knowledge Continuum,* http://choo.fis.utoronto.ca/Kluwer/DataInfoKnow.html, Visited Ocotber 7th, 2002.

[cIDf 2002] Content ID Forum, cIDf, 2002, *cIDf Specification 1.1,* September 11th, 2002. See also http://www.cidf.org/english/documents/cIDfSpecV11R11E.pdf.

[Citeseer 2002] The NEC Research Institute, 2002, *The NEC Research Institute Scientific Literature Digital Library,* http://citeseer.nj.nec.com/, Visited May 14th, 2003.

[Clarke 2001] Clarke, P., EE Times, 2001, *Sanyo Licenses Hantro MPEG-4 Core for PDA Device,* http://www.commsdesign.com/story/OEG20010920S0063, Visited February 6th, 2003.

[Currier 1995] Currier, B., 1995, *Digital Video Codec Choices, Part 2,* http://www.synthetic-ap.com/qt/codec2.html, Visited February 6th, 2003.

[Cyc 2002] Cycorp, Inc., 2002, *Cycorp: Makers of the Cyc Knowledge Server for Artificial Intelligence-Based Common Sense,* http://www.cyc.com/, Visited October 14th, 2002.

[Dallermassl and Helic 2003] Dallermassl, C., and Helic, D., 2003, *Skript zur Vorlesung Multimediale Informationssysteme,* ftp://courses.iicm.edu/courses/mmis/script_main.pdf, Visited April 1st, 2003.

[Davenport 2002] Davenport, K., 2002, *Data, Information, Knowledge Hierarchy,* Library Journal, Volume 127, Number 8, May 2002, page 10.

[Davis 2001] Davis, L., 2001, *Profiting from Innovations in Digital Information Goods: the Role of Intellectual Property Rights,* Proceedings of the IEEE International Conference on Management of Engineering and Technology (PICMET'01), 2001, p. 78.

[Day 2001] Day, N., 2001, *MPEG-7 Applications: Multimedia Search and Retrieval.* See also http://www.carc.aist.go.jp/nlwww/mmac/resources/pdf/day.pdf.

[DCMES 1999] Dublin Core Metadata Initiative, 1999, *Dublin Core Metadata Element Set, Version 1.1: Reference Description.* See also http://dublincore.org/documents/1999/07/02/dces.
[DCMI 2002] Dublin Core Metadata Initiative, 2002, *Dublin Core Metadata Initiative,* http://dublincore.org/, Visited November 15th, 2002.
[DeLay 2001] DeLay, J. L., 2001, *Broadband Data Delivery over Television Networks,* Proceedings of the 2001 Second International Workshop on Digital and Computational Video, 2001, pp. 36-43.
[Delbrouck 2002] Delbrouck, D., ZDNet.de, 2002, *MPEG 4-Player für Linux-PDA,* http://news.zdnet.de/story/0,,t101-s2120174,00.html, Visited February 6th, 2003.
[DeMet 2001] DeMet, G. D., 2001, http://www.palantir.net/cgi-bin/image.cgi?pics/station04.jpg, Visited February 8th, 2003.
[Dieng-Kuntz and Matta 2002] Dieng-Kuntz, R., and Matta, N. (eds.), 2002, *Knowledge Management and Organizational Memories,* Kluwer Academic Publishers, Boston 2002.
[Dietinger and Maurer 1998] Dietinger, T., and Maurer, H., 1998, *GENTLE – GEneral Networked Training and Learning Environment,* Proceedings of the EDMedia'98 Conference, 1998. See also http://www.iicm.edu/liberation/iicm_papers/edmedia98.pdf.
[DOI 2003] International Digital Object Identifier Foundation, 2003, *The DOI® Handbook Version 2.6.0, January 2003.* See also http://dx.doi.org/10.1000/186.
[Dourish et al. 2000] Dourish, P., Edwards, W. K., Howell, J., et al., 2000, *A Programming Model for Active Documents,* Proceedings of the 2000 ACM Symposium on User Interface Software and Technology (UIST2000), San Diego, California, November 2000. See also http://delivery.acm.org/10.1145/360000/354410/p41-dourish.pdf.
[DSM-CC 1997] DSM-CC Sub-Group, International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 1997, *DSM-CC FAQ Version 1.0,* http://mpeg.telecomitalialab.com/faq/mp2-dsm/mp2-dsm.htm, Visisted January 25th, 2003.
[Dufaux et al. 2000] Dufaux, A., Besacier, L., Ansorge, M., and Pellandini, F., 2000, *Automatic Sound Detection and Recognition for Noisy Environment,* http://herakles.imag.fr/besacier/Publis/eusipco2000-2.pdf, Visited April 25th, 2003.
[DVB 2002] The DVB Project Office, 2002, *iTV Standards Adoption with TV Operators,* http://www.dvb.org/images/DVB_penetration_maps_DVB-MHP.pdf, Visited February 5th, 2003.
[ELC 2003] Embedded Linux Consortium, 2003, *ELC – Embedded Linux Consortium,* http://www.embedded-linux.org/, Visited January 27th, 2003.
[eLS 2003] Hyperwave AG, 2003, *Hyperwave – Hyperwave eLearning Suite,* http://www.hyperwave.com/e/products/els.html, Visited March 18th, 2003.
[Erickson 2001] Erickson, J. S., 2001, *Information Objects and Rights Management: A Mediation-Based Approach to DRM Interoperability,* D-Lib Magazine, Volume 7, Number 4, April 2001. See also http://www.dlib.org/dlib/april01/erickson/04erickson.html.
[Erickson 2002] Erickson, J. S., 2002, *Digital Rights Management: Business and Technology,* D-Lib Magazine, Volume 8, Number 2, February 2002. See also http://www.dlib.org/dlib/february02/02bookreview.html.
[Fabri et al. 1999] Fabri, S., Cellatoglu, A., and Kondoz, A., 2001, *Transmission of Multimedia Services over GPRS using MPEG-4 Coded Video,* Proceedings of the 1999 IEEE VTC, Amsterdam, pp. 401-405. See also http://www.ee.surrey.ac.uk/Personal/S.Fabri/publications_files/vtc99_vid_gprs.pdf.
[Fairhurst 2001a] Fairhurst, G., 2001, *MPEG-2,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/mpeg2.html, Visited January 21st, 2003.
[Fairhurst 2001b] Fairhurst, G., 2001, *MPEG-2 Transmission,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/mpeg2-trans.html, Visited January 21st, 2003.
[Fairhurst 2001c] Fairhurst, G., 2001, *MPEG-2 and DVB Standards,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/mpeg2-docs.html, Visited January 21st, 2003.
[Fairhurst 2001d] Fairhurst, G., 2001, *DVB: Digital Video Broadcasting,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/dvb.html, Visited January 21st, 2003.
[Fairhurst 2001e] Fairhurst, G., 2001, *DVB Transmission,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/dvb-trans.html, Visited January 21st, 2003.

[Fairhurst 2001f] Fairhurst, G., 2001, *Data Transmission Using MPEG-2 and DVB,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/dsm-cc.html, Visited January 22nd, 2003.

[Fairhurst 2001g] Fairhurst, G., 2001, *Broadband Multimedia Satellite Systems,* http://www.erg.abdn.ac.uk/public_html/research/future-net/digital-video/bband-sat.html, Visited January 22nd, 2003.

[Feiss 2002] Apple Computer, Inc., 2002, *Apple – Switch – Ads – Ellen Feiss,* http://www.apple.com/switch/ads/ellenfeiss.html, Visited November 28th, 2002.

[Forbes 2001] Forbes, G., 2001, *Closed Captioning Transmission and Display in Digital Television,* Proceedings of the Second International Workshop on Digital and Computational Video, 2001, pp. 126-131.

[Fowler 2002] Fowler, T. B., 2002, *Technology's Changing Role in Intellectual Property Rights,* IT Professional, Volume 4, Issue 2, March-April 2002, pp. 39-44.

[FreeSpeech 2003] Valin, J.-M., 2003, *Open Mind Speech – Free Speech Recognition for Linux,* http://freespeech.sourceforge.net/, Visited April 25th, 2003.

[Garfield 1955] Garfield, E., 1955, *Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas,* Science, Volume 122, Number 3159, July 1955, pp. 108-111. See also http://www.garfield.library.upenn.edu/papers/science_v122(3159)p108y1955.html.

[Gentle 2002] Institute for Information Processing and Computer-Supported New Media (IICM), 2002, *Gentle-WBT – Web-based Training in a New Dimension,* http://www.gentle-wbt.com/, Visited March 18th, 2003.

[Gereöffy 2002] Gereöffy, A., 2002, *Introduction of MPlayer,* http://www.mplayerhq.hu/homepage/essays/konf2002-arpi-en.html, Visited February 27th, 2003.

[Guetl and Maurer 2000] Guetl, C., and Maurer, H., 2000, *Kurzfassung für den Forschungsauftrag 61090: Intelligente Wissenserfasung und Wiederauffindung in künftigen WWW-Systemen. Forschungsprojekt im Bereich WWW-Wissensorganisation.* See also http://www2.iicm.edu/cguetl/papers/km_www/kurzfassungv1_0.pdf.

[Haas and Mayer 2001] Haas, W., Mayer, H., 2001, *MPEG and its Relevance for Content-based Multimedia Retrieval,* Journal of Universal Computer Science, Volume 7, Issue 6, June 2001, pp. 530-547. See also http://www.jucs.org/jucs_7_6/mpeg_and_its_relevance/.

[Hardjono and Baugher 2001] Hardjono, T., and Baugher, M., 2001, *Internet Digital Rights Management (IDRM),* IETF DRM Meeting (IETF50), March 2001. See also http://www.idrm.org/idrm1_ThomasHardjono.pdf.

[Hasson 2002] Hasson, J., University of Connecticut Health Center, *Atrial Septal Defect of the Myocardium,* http://radiology.uchc.edu/eAtlas/CV/767.htm, Visited November 17th, 2002.

[HAVi 2002] TwonkyVision, 2002, *HAVi UI for MHP 1.0.2,* http://www.twonkyvision.de/products/haviui.html, Visited January 26th, 2003.

[Heinrich and Maurer 2000] Heinrich, E., Maurer, H., 2000, *Active Documents: Concept, Implementation and Applications,* Journal of Universal Computer Science, Volume 6, Issue 12, December 2000, pp. 1197-1202. See also http://www.jucs.org/jucs_6_12/active_documents_concept_implementation/.

[Heinrich et al. 2001] Heinrich, E., Johnson, R., Luo, D., Maurer, H., and Sapper, M., 2003, *Learner-Formulated Questions in Technology-Supported Learning Applications,* Proceedings of ED-MEDIA 2001, Tampere, pp. 720-725. See also http://www.iicm.edu/iicm_papers/learner_formulated_questions.pdf.

[Herbst and Huebner 2002] Herbst, H., and Huebner, J.-H., 2002, *Pathologie-Online – Vorwort,* http://www.pathologie-online.de/vw/index.htm, Visited April 15th, 2003.

[Herrera-Boyer 2002] Boyer-Herrera, P., 2002, *Setting Up an Audio Database for Music Information Retrieval Benchmarking,* http://music-ir.org/evaluation/wp2/wp2_herrera.pdf, Visited April 10th, 2003.

[Herzog 1994] Herzog, M., 1994, *Case-Based Reasoning,* http://www.dbai.tuwien.ac.at/staff/herzog/thesis/node14.html, Visited October 23rd, 2002.

[Hicks 2001] Hicks, D., 2001, *From Personal Digital Libraries to Knowledge Management,* Proceedings of the I-Know '01 Conference, Graz, Austria, 2001. See also http://www.know-center.at/en/conference/i-know01/pdf/dhicks.pdf.

[Hicks and Tochtermann 2001] Hicks, D., and Tochtermann, K., 2001, *Personal Digital Libraries and Knowledge Management,* Journal of Universal Computer Science, Volume 7, Issue 7, July 2001, pp. 550-565. See also http://www.jucs.org/jucs_7_7/personal_digital_libraries_and/.

[Hierl et al. 2003] Hierl, A., Fischer, M., et al., 2003, *Reformprojekte in der Medizin – Synergie durch Überblick,* http://cbt.medinn.med.uni-muenchen.de/levacmed/home/default.asp, Visited April 12th, 2003.

[Hillmann 2001] Hillmann, D., 2001, *Using Dublin Core,* http://dublincore.org/documents/2001/04/12/usageguide/, Visited September 30th, 2002.
[Hitchcock et al. 2002] Hitchcock, S., Bergmark, D., Brody, T., et al., 2002, *Open Citation Linking. The Way Forward,* D-Lib Magazine, Volume 8, Number 10, October 2002. See also http://www.dlib.org/dlib/october02/hitchcock/10hitchcock.html.
[Hobbs and Swain 2001] Hobbs, M. L., and Swain, P., Radio Spectrum and Broadcasting Policy Group, 2001, *Digital Television,* Discussion Paper 248134-1, December 2001. See also http://www.executive.govt.nz/minister/hobbs/digital/digital.pdf.
[Hunter 2001] Hunter, J., 2001, *MPEG-7: Transforming Digital Video Asset Description.*
[Hunter 2002] Hunter, J., 2002, *An Application Profile which combines Dublin Core and MPEG-7 Metadata Terms for Simple Video Description,* http://www.metadata.net/harmony/video_appln_profile.html, Visited March 2nd, 2003.
[Hunter and Newmarch 1999] Hunter, J., and Newmarch, J., 1999, *An Indexing, Browsing, Search and Retrieval System for Audiovisual Libraries.* In: Abiteboul, A., and Vercoustre, A.-M. (eds.), *ECDL '99, LNCS 1696,* pp. 76-91, Springer-Verlag Berlin, 1999.
[Hyperwave 2003] Hyperwave AG, 2003, *Hyperwave – The Power of Wisdom,* http://www.hyperwave.com/, Visited March 18th, 2003.
[Iannella 2001] Iannella, R., 2001, *Digital Rights Management (DRM) Architectures,* D-Lib Magazine, Volume 7, Number 6, June 2001. See also http://www.dlib.org/dlib/june01/iannella/06iannella.html.
[Iannella 2002] Iannella, R., 2002, *ODRL – The Language for DRMS,* Digital Rights Management Systems (DRMS) and Public Interests Symposium, Sydney, Australia, March 2002. See also http://www.bakercyberlawcentre.org/2002/DRMS_Papers/DRMS-Sym-RI-Mar2002.pdf.
[IBM 2001] IBM Haifa Research Lab, 2001, *IBM Demonstrates MPEG-4 Java Technology,* http://www.haifa.il.ibm.com/info/feature_ibm_haifa_hammers.html, Visited March 8th, 2003.
[IBM 2002] IBM alphaWorks, 2002, *MPEG-4 Video for JMF,* http://www.alphaworks.ibm.com/tech/mpeg-4/, Visited March 8th, 2003.
[IDRM 2001] N.n., 2001, *Possible IDRM Study Items,* http://www.idrm.org/idrm1_IDRMWorkItems.pdf, Visited January 17th, 2002.
[IDRM 2002] Internet Digital Rights Management, IETF, 2002, *Internet Digital Rights Management,* http://www.idrm.org/, Visited November 9th, 2002.
[IMDb 2002] Internet Movie Database, Inc., 2002, *The Internet Movie Database (IMDb),* http://www.imdb.com/, Visisted November 28th, 2002.
[IMKA 2001] Benitez, A. B., 2001, *IMKA Project,* http://ana.ctr.columbia.edu/imka/, Visited November 14th, 2002.
[Infineon 2001] Infineon Technolgies, 2001, *Infineon and Toshiba announce interface compatibility between the companies' respective UMTS/GSM baseband and MPEG-4 video solutions for upcoming 3rd Generation cellular terminals,* http://www.infineon.com/news/press/102_041e.htm, Visited February 6th, 2003.
[ISI 2003] Thomson ISI , 2003, *Scientific Citation Index Expanded,* http://www.isinet.com/isi/products/citation/scie/index.html, Visited May 14th, 2003.
[Java-TV 2002a] Sun Microsystems, Inc., 2002, *Java™ Technology in Digital TV,* http://java.sun.com/products/javatv/, Visited January 22nd, 2003.
[Java-TV 2002b] Sun Microsystems, Inc., 2002, *The Java TV™ API Overview: Setting the Standard for a Digital Future,* http://java.sun.com/products/javatv/overview.html, Visited January 22nd, 2003.
[Java-WDS 2003] Sun Microsystems, Inc., 2003, *Java Wireless Device Solution Packages,* http://wwws.sun.com/software/jpe/wireless/, Visited February 27th, 2003.
[Java-WTK 2002] Sun Microsystems, Inc., 2002, *Java™ 2 Platform Micro Edition Wireless Toolkit version 1.0.4-01 for Windows,* http://java.sun.com/products/j2mewtoolkit/download.html, Visited February 27th, 2003.
[Java-WTK 2003] Sun Microsystems, Inc., 2003, *Java™ 2 Platform Micro Edition, Wireless Toolkit,* http://java.sun.com/products/j2mewtoolkit/, Visited February 27th, 2003.
[JMF 2002a] Sun Microsystems, Inc., 2002, *Java™ Media Framework API,* http://java.sun.com/products/java-media/jmf/, Visited January 26th, 2003.
[JMF 2002b] Sun Microsystems, Inc., 2002, *JMF 2.1.1 Software Documentation,* http://java.sun.com/products/java-media/jmf/2.1.1./documentation.html, Visited January 26th, 2003.

[JUCS 2003] Journal of Universal Computer Science (J.UCS), 2003, *Welcome to the Journal of Universal Computer Science,* http://www.jucs.org/, Visited March 18th, 2003.
[Karsten and Neumann 2002] Karsten, G., and Neumann, O., 2002, *Einsatz der Lehr- und Lernplattform JaTeK in der Medizinausbildung: Erste Erfahrungen aus der Praxis,* Project Workshop "e-Learning in Medical Education", Bochum, Germany, November 2002. See also http://ms2.physiologie.uni-kiel.de/medu/pdf/karsten.pdf.
[Kasenna 2002] Kasenna, Inc., 2002, *Products,* http://www.kasenna.com/products/mediabase/index.jsp, Visited February 6th, 2003.
[Keller 2001] Keller, P. E., 2001, *Neural Networks: What are Neural Networks?,* http://www.emsl.pnl.gov:2080/proj/neuron/neural/what.html, Visited October 29th, 2002.
[Khattak and Kanafani 1996] Khattak, A., and Kanafani, A., 1996, *Case-Based Reasoning: A Planning Tool for Intelligent Transportation Systems,* Transportation Research, Part C: Emerging Technologies, Volume 4, Number 5, pp. 267-288. See also http://www.unc.edu/~khattak/cbr.html.
[KiSS 2002] KiSS Technology, 2002, *KiSS DVD Player DP-450,* http://www.kiss-technology.com/projects/dvd_450.pdf, Visited January 27th, 2003.
[Knight 2003] Knight, W., 2003, *Copy Protection is Crossing Boundaries,* http://www.newscientist.com/news/news.jsp?id=ns99993250, Visited January 10th, 2003.
[Koenen 2001a] Koenen, R., International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2001, *Intellectual Property Management and Protection in MPEG Standards,* http://mpeg.telecomitalialab.com/standards/ipmp/index.htm, Visisted November 8th, 2002.
[Koenen 2001b] Koenen, R., International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2001, *From MPEG-1 to MPEG-21: Creating an Interoperable Multimedia Infrastructure,* http://mpeg.telecomitalialab.com/documents/from_mpeg-1_to_mpeg-21.htm, Visited November 7th, 2002.
[Koenen 2001c] Koenen, R., 2001, *Intellectual Property Management and Protection in MPEG,* IETF DRM Meeting (IETF50), March 2001. See also http://www.idrm.org/idrm1_Koenen.pdf.
[Köhler et al. 2001] Köhler, J., Biatov, K., Larson, M., et al., 2001, *AGMA Automatic Generation of Meta Data for Audio-Visual Content in the Context of MPEG-7.* In: Fleischmann, M. (ed.), 2001, *Cast01 – Living in Mixed Realities.*
[Krottmaier 2002] Krottmaier, H., 2002, *Aspects of Modern Publishing Systems,* PhD Thesis. See also http://www.iicm.edu/thesis/hkrott_diss.pdf.
[Krottmaier and Maurer 2001] Krottmaier, H., and Maurer, H., 2001, *Transclusions in the 21st Century,* Journal of Universal Computer Science, Volume 7, Issue 12, December 2001, pp. 1125-1136. See also http://www.jucs.org/jucs_7_12/transclusions_in_the_21st/.
[Kulper 2002] Kulper, S., 2002, *Bauhaus Dessau, Walter Gropius 1926,* http://revolution.mit.edu/courses/ju/4.206/web/bauhaus-plan.gif, Visited April 24th, 2003.
[Langkafel 2002a] Langkafel, P., 2002, *E-Learning in der Medizin. Synergien und state of the art,* http://www.e-learning-in-der-medizin.de/, Visited April 12th, 2003.
[Langkafel 2002b] Langkafel, P., 2002, *E-Learning in der Medizin. Synergien und state of the art. Konzept,* http://www.e-learning-in-der-medizin.de/konzept.htm, Visited April 12th, 2003.
[Lawrence et al. 1999] Lawrence, S., Giles, C. L., and Bollacker, K., 1999, *Digital Libraries and Autonomous Citation Indexing,* IEEE Computer, Volume 32, Number 6, June 1999, pp. 67-71. See also http://www.neci.nec.com/~lawrence/papers/aci-computer98/aci-computer99.pdf.
[Leake 1996] Leake, D. B., 1996, *CBR in Context: the present and Future.* In: Leake, D. B. (ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions,* pp. 1-35, AAAI Press, Menlo, CA, USA, 1996.
[Lee 2002] Lee, D., 2002, *A Lesson in Lessons Learned,* http://appl.nasa.gov/knowledge/issues/lessons_learned.html, Visited October 22nd, 2002.
[Lee and Hyun 2002] Lee, S.-M., and Hyun, S. J., 2002, *MPEG-7 Database Team,* http://swing.icu.ac.kr/material/Mpeg/MPEG-7%20Database%20Team.ppt, Visited February 27th, 2003.
[Lemstrom and Perttu 2000] Lemstrom, K., and Perttu, S., 2000, *SEMEX – An Efficient Music Retrieval Prototype,* International Symposium on Music and Information Retrieval, Plymouth, Massachusetts, October 2000. See also http://ciir.cs.umass.edu/music2000/papers/lemstrom_paper.pdf.

[Lennon and Maurer 1994] Lennon, J., and Maurer, H., 1994, *Applications and Impact of Hypermedia Systems: An Overview,* Journal of Universal Computer Science, Volume 0, Issue 0, pp. 54-107. See also http://www.jucs.org/jucs_0_0/applications_and_impact_of/html/paper.html.

[Lenz 1998] Lenz, M., et al. (eds.), 1998, *Case-Based Reasoning Technology. From Foundations to Applications.* Springer-Verlag Berlin, 1998.

[Liebowitz 2001] Liebowitz, J., 2001, *Knowledge Management and Its Link to Artificial Intelligence,* Expert Systems with Applications, Volume 20, Number 1, January 2001, pp 1-6.

[Liebowitz 2002] Liebowitz, J., 2002, *A Look at NASA Goddard Space Flight Center's Knowledge Management Initiatives,* IEEE Software, Volume 19, Issue 3, May-June 2002, pp. 40-42.

[Ligos 2003] Ligos Corporation, 2003, *MediaRig™ Core. A Powerful Encoding and Transcoding Solution,* http://www.ligos.com/pdf_docs/mediarigcore.pdf, Visited April 10th, 2003.

[LLIS 2002] Smith, E. M., et al., 2002, *Nasa Lessons Learned,* http://llis.nasa.gov/, Visited October 10th, 2002.

[Lugmayr et al. 2002] Lugmayr, A., Niiranen, S., Mailaparampil, A., Rautavirta, P., et al., 2002, *Applying MPEG-21 in Digital Television – Example Use Scenarios: ePostcard, eGame, and eTicket,* Proceedings of the IEEE International Conference on Multimedia and Expo, 2002, Volume 2, pp. 293-296.

[Lustrup 2002] Lustrup, B., KiSS Technology, 2002, *New DVD Player Receives Digital Media Directly from the Internet,* http://www.sigmadesigns.com/news/press_releases/pr_pdf/021127.pdf, Visited November 29th, 2002.

[Malhotra 2001] Malhotra, Y., 2001, *Expert Systems for Knowledge Management: Crossing the Chasm Between Information Processing and Sense Making,* Expert Systems with Applications, Volume 20, Number 1, January 2001, pp 7-16.

[Martinek 1999] Martinek, K., 1999, *Professional Digital Video Tape Recorder Comparison,* http://kensystem.com/kensys/vtr.htm, Visited April 6th, 2003.

[Maurer 2001] Maurer, H., 2001, *Beyond Classical Digital Libraries,* Proceedings of the NIT Conference, Global Digtial Library Development in the New Millenium, Beijing, Tsinghua University Press, 2001, pp. 165-173. See also http://www.iicm.edu/iicm_papers/beyond_digital_libraries.doc.

[Maurer 2003] Maurer, H., 2003, *Important Aspects of Knowledge Management,* LNCS 2598, Special Issue, Springer Verlag, 2003, pp. 245-254.

[Maurer and Guetl 2000] Maurer, H., and Guetl, C., 2000, *Intelligente Wissenserfassung und Wiederauffindung in künftigen WWW-Systemen,* http://www2.iicm.edu/cguetl/papers/km_www/wissen_end_v08.pdf, Visited April 15th, 2003.

[Maurer and Oliver 2003] Maurer, H., and Oliver, R., 2003, *The Future of PCs and Implications on Society.*

[Maurer and Tochtermann 2002] Maurer, H., Tochtermann, K., 2002, *On a New Powerful Model for Knowledge Management and its Applications,* Journal of Universal Computer Science, Volume 8, Issue 1, January 2002, pp. 85-96. See also http://www.jucs.org/jucs_8_1/on_a_new_powerful/.

[Mediabase 2002] Kasenna, Inc., 2002, *Kasenna™ MediaBase XMP,* http://www.kasenna.com/products/mediabase/xmp.jsp, Visited February 6th, 2003.

[Medianovo 2003] Leichsenring, M., 2003, *Medianovo – die medizinische Mediendatenbank im Web,* http://www.medianovo.com/media_67/info/start.asp, Visited April 12th, 2003.

[Meier 2002] Meier, W., 2002, *eXist: An Open Source Native XML Database,* http://exist-db.org/webdb.pdf, Visited February 27th, 2003.

[Meier 2003] Meier, W., 2003, *eXist: Open Source XML Database,* http://exist-db.org/, Visited February 27th, 2003.

[Meyer and Fontaine 2000] Meyer, L., and Fontaine, G., Institute de l'audiovisuel et des télécommunications en europe, 2000, Development of Digital Television in the European Union, Reference Report 1999.

[MHP 2002a] The DVB Multimedia Home Platform (DVB-MHP), 2002, *What is MHP? Overview,* http://www.mhp.org/what_is_mhp/overview.html, Visited December 10th, 2002.

[MHP 2002b] The DVB Multimedia Home Platform (DVB-MHP), 2002, *Technical Essentials,* http://www.mhp.org/technical_essen/tech_essentials.html, Visited December 10th, 2002.

[MHP-Forum 2002] MHP-Forum, 2002, *MHP Fragen und Antworten,* http://www.gruen-online.de/Frontpage/content/wissen/faq.htm, Visited January 22nd, 2003.

[Miller et al. 1993] Miller, G. A., Beckwith, R., Fellbaum C., et al., 1993, *Introduction to WordNet: An On-line Lexical Database.* See also http://www.cogsci.princeton.edu/~wn/5papers.pdf.

[Miron et al. 2001] Miron, M., DeMartini, T., Wang, X., and Gandee, B., ContentGuard, 2001, *XrML – The Language for Digital Rights™,* MPEG Workshop, Pattaya, Thailand, December 2001. See also http://www.xrml.org/reference/MPEG_Thailand.pdf.

[Mooney 2001] Mooney, S. P., 2001, *Interoperatbility: Digital Rights Management and the Emerging EBook Environment,* D-Lib Magazine, Volume 7, Number 1, January 2001. See also http://www.dlib.org/dlib/january01/mooney/01mooney.html.

[Moreno et al. 2002] Moreno, P. J., Van Thong, J.-M., et al., 2002, *From Multimedia Retrieval to Knowledge Management,* Computer, Volume 35, Issue 3, March 2002, pp 58-66.

[Morris 2002a] Morris, S., 2002, *An Introduction to Digital TV and DCB Technologies,* http://www.mhp-interactive.org/dtv-intro/index.html, Visited December 10th, 2002.

[Morris 2002b] Morris, S., 2002, *A Little More about DVB,* http://www.mhp-interactive.org/dvb-background.html, Visited December 10th, 2002.

[Morris 2002c] Morris, S., 2002, *An Introduction to the DVB Multimedia Home Platform,* http://www.mhp-interactive.org/presentations/MHP-intro.ppt, Visited January 23rd, 2003.

[MPEG-1 1996] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 1996, *MPEG-1: Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s. Short MPEG-1 Overview,* http://mpeg.telecomitalialab.com/standards/mpeg-1/mpeg-1.htm, Visited November 7th, 2002.

[MPEG-2 2000] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2000, *MPEG-2: Generic Coding of Moving Pictures and Associated Audio Information. Short MPEG-2 Description,* http://mpeg.telecomitalialab.com/standards/mpeg-2/mpeg-2.htm, Visited November 7th, 2002.

[MPEG-4 2002] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2002, *Overview of the MPEG-4 Standard,* http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm, Visited November 7th, 2002.

[MPEG-7 2001] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2001, *Introduction to MPEG-7 (v3.0),* http://ipsi.fhg.de/delite/Projects/MPEG7/Documents/W4325%20M7%20Intro.htm, Visited November 11th, 2002.

[MPEG-7 2002] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2002, *MPEG-7 Overview (version 8),* http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm, Visited November 7th, 2002.

[MPEG-21 2002] International Organisation for Standardisation ISO/IEC JTC1/SC29/WG11, 2002, *MPEG-21 Overview v.5,* http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm, Visited November 7th, 2002.

[MPEG-IMX 2003] Sony Electronics, Inc., 2002, *Sony MPEG IMX – MPEG Technology Online Resources,* http://www.sonympeg-imx.com/, Visited March 3rd, 2003.

[Mülner 2001] Mülner, H., 2001, *Software für ein multimediales Lexikon,* Presentation.

[MSAR 2003a] The Microsoft Agent Ring, http://www.msagentring.org/chars/images/claude.gif, Visited February 9th, 2003.

[MSAR 2003b] The Microsoft Agent Ring, http://www.msagentring.org/chars/images/merlin.gif, Visited February 9th, 2003.

[MSAR 2003c] The Microsoft Agent Ring, http://www.msagentring.org/chars/images/max.gif, Visited February 9th, 2003.

[Nakamura and Asada 1995] Nakamura, T., and Asada, M., 1995, *Motion Sketch : Acquisition of Visual Motion Guided Behaviors,* Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), pp. 126-132. See also http://www_robotics.ccm.eng.osaka-u.ac.jp/user/papers/1995/Nakamura95d.pdf.

[Nelson 1981] Nelson, T. H., 1981, *Literary Machines,* Mindfull Press, 1981.

[Nelson 1996] Nelson, T. H., 1996, *Generalized Links, Micropayment and Transcopyright,* http://www.almaden.ibm.com/almaden/npuc97/1996/tnelson.htm, Visited May 1st, 2003.

[NNLS 2002] Navy Welfare Development Command, 2002, *Navy Lessons Learned System: Overview,* http://www.nwdc.navy.mil/nlls/nllsoverview.asp, Visited October 18th, 2002.

[OAI 2002] Lagoze, C., Van de Sompel, H., Nelson, M., and Warner, S. (eds.), Open Archives Initiative, 2002, *The Open Archives Intitiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 2002-06-14,* http://www.openarchives.org/OAI/openarchivesprotocol.html, Visited November 15th, 2002.

[OAI 2003] Open Archives Initiative, 2003, *Registered Data Providers,* http://www.openarchives.org/Register/BrowseSites.pl, Visited May 23rd, 2003.

[ODRL 2002a] Iannella, R., ORDL, 2002, *Open Digital Rights Language (ORDL). Version 1.1.* See also http://www.odrl.net/1.1/ODRL-11.pdf.
[ODRL 2002b] Iannella, R., ODRL, 2002, *Open Digital Rights Language. A Rights Expression Language for Digital Asset Management and E-Commerce,* http://odrl.net/docs/ODRL-brochure.pdf, Visited January 17th, 2003.
[O'Hagan 2001] O'Hagan, C., 2001, *The Integration of Television and the Internet,* Proceedings of the 2001 IEEE International Conference on Advanced Learning Technologies, pp. 475-477.
[Omras 2000] Digital Libraries Initiative 2, National Science Foundation, 2000, *Online Music Recognition and Searching (OMRAS),* http://www.dli2.nsf.gov/internationalprojects/JISC/OMRAS/Full_desc.html, Visited April 25th, 2003.
[OpenCyc 2002] Cycorp, Inc., 2002, *OpenCyc.org: Formalized Common Knowledge,* http://www.opencyc.org/, Visited October 14th, 2002.
[Padmanabhan et al. 2001] Padmanabhan, M., Saon, G., et al., 2001, *Evolution of the Performance of Automatic Speech Recognition Algorithms in Transcribing Conversational Telephone Speech,* Proceedings of the 2001 IEEE Conference on Instrumentation and Measurement Technology, Budapest, Hungary, 2001. See also http://www.research.ibm.com/voicemail/pdf/imtc2001.pdf.
[Paskin 2002a] Paskin, N., International DOI Foundation, 2002, *DOI Overview,* http://dx.doi.org/10.1000/202, Visited January 17th, 2003.
[Paskin 2002b] Paskin, N., International DOI Foundation, 2002, *Digital Object Identifier: Overview,* http://dx.doi.org/10.1000/210, Visited January 17th, 2003.
[Paskin 2002c] Paskin, N., International DOI Foundation, 2002, *Digital Object Identifier,* http://dx.doi.org/10.1000/237, Visited January 17th, 2003.
[Paxal 2001] Paxal, V., Telenor R&D, 2001, *DVB with Return Channel via Satellite,* http://www.dvb.org/dvb_technology/whitepaper-pdf-docs/DVBrcspresentation.pdf, Visited January 23rd, 2003.
[Peek 2002] Peek, T., 2002, *DVB Today and Tomorrow: Convergence Issues,* Innovative Consumer Electronics 2002, http://www.gfu.de/pages/download_files/DVB_Convergence.ppt, Visited January 21st, 2003.
[Perkis 2001] Perkis, A., 2001, MPEG-4 *Media Resource Delivery,* http://www.tele.ntnu.no/users/andrew/Papers/FFV202/DMIF-streaming(4perpage).pdf, Visited February 25th, 2003.
[Photobook 1994] Pentland, A., Picard, R. W., and Sclaroff, S., 1994, *Photobook: Content-Based Manipulation of Image Databases,* IEEE Multimedia, Summer 1994, pp. 73-75. See also http://web.media.mit.edu/~tpminka/photobook/.
[Piesing and Löytänä 1999] Piesing, J., and Löytänä, L., 1999, *The DAVIC APIs for Interactive TV,* http://www.davic.org/Info_Day/piesing.pdf, Visited January 26th, 2003.
[Piva et al. 2002] Piva, A., Bartolini, F., and Barni, M., 2002, *Managing Copyright in Open Networks,* IEEE Internet Computing, Volume 6, Issue 3, May-June 2002, pp. 18-26.
[Polyphonic 2003] Polyphonic HMI, 2003, *Polyphonic HMI Human Media Interface,* http://www.polyphonichmi.com/, Visited February 18th, 2003.
[Powell 2002] Powell, A., 2002, *Guidelines for Implementing Dublin Core in XML,* http://dublincore.org/documents/2002/09/09/dc-xml-guidelines/, Visited September 30th, 2002.
[QTVR 2002] Apple Computer, Inc., 2002, *Inside QuickTime: Virtual Reality.* See also http://developer.apple.com/techpubs/quicktime/qtdevdocs/PDF/insideqt_qtvr.pdf.
[Quicktime 2002] Apple Computers, Inc., 2002, *Apple – Quicktime,* http://www.apple.com/quicktime/, Visited November 10th, 2002.
[Rein 2002] Rein, L., 2002, *Comparing Rights Languages,* http://www.lisarein.com/rightslang.html, Visited January 18th, 2003.
[RFC-2413 1998] Weibel, S., Kunze, J., Lagoze, C., and Wolf, M., 1998, *Request for Comments: 2413. Dublin Core Metadata for Resource Discovery,* http://www.ietf.org/rfc/rfc2413.txt, Visited September 30th, 2002.
[Rogers et al. 2001] Rogers, D., Hunter, J., and Kosovic, D., 2001, *The TV-Trawler Project.* See also http://archive.dstc.edu.au/RDU/staff/jane-hunter/tv-trawler.pdf.
[Rose et al. 1999] Rose, M., Dormann, C., Olesen, H., et al., 1999, *White Paper on Interactive Television,* http://www.cti.dtu.dk/~berco/html/cti/research/ITV%20white%20paper.html, Visited April 2nd, 2003.
[Rust and Bide 2000] Rust, G., and Bide, M., 2000, *The <indecs> Metadata Framework: Principles, Model and Data Dictionary,* http://www.indecs.org/pdf/framework.pdf, Visited January 18th, 2003.

[Rutkowski 2002] Rutkowski, K., 2002, *Siemens Mobile Phone Products for 2003. The Brill Report,* http://www.tcf.net/siemens.html, Visited February 25th, 2003.

[Sakamoto 2001] Sakamoto, H., Content ID Forum, 2001, *Overview of the Content ID Forum.* See also http://www.cidf.org/japanese/english/docs/gen/cidf-gen-en-13.pdf.

[Sakamoto 2002] Sakamoto, H., Content ID Forum, 2002, *Introduction of MPEG-21: Activities and DII&D.* See also http://www.cidf.org/japanese/english/docs/gen/cidf-gen-en-76.pdf.

[Scalise 2001] Scalise, F., 2001, *DVB-RCT Return Channel Terrestrial,* http://www.broadcastpapers.com/broadbiz/RCT%20white%20paper_final.doc, Visited January 23rd, 2003.

[Sigma 2002] Sigma Designs, Inc., 2002, *EM8605: HDTV Decoder with MPEG-4 for Set-top Appliances,* http://www.sigmadesigns.com/pdf_docs/em8605_brochure.pdf, Visited January 27th, 2003.

[Sigma 2003] Sigma Designs, Inc., 2003, *Media Processing for Consumer Appliances,* http://www.sigmadesigns.com/, Visited January 27th, 2003.

[Smith-Chaigneau 2001] Smith-Chaigneau, A., DVB Project Office, 2001, *Multimedia Home Platform. DVB-MHP – A Snapshot – October 2001.*

[Sony 1999] Sony Electronics, Inc., 1999, *The MPEG Top Ten Countdown. Answers to the Top Ten Questions about MPEG Technology.* See also http://www.sonympeg-imx.com/resources/mpegtop10.pdf.

[Sphinx 2002] The Sphinx Group, Carnegie Mellon University, 2002, *CMU Sphinx. The CMU Sphinx Group Open Source Speech Recognition Engines,* http://www.speech.cs.cmu.edu/sphinx/index.html, Visited April 25th, 2003.

[SRecMgr 1997] Apple Computer, Inc., 1997, *Inside Macintosh: Speech Recognition Manager,* http://developer.apple.com/techpubs/mac/speechrecogmgr/srec-2.html, Visited April 25th, 2003.

[SRecMgr 2003] Apple Computer, Inc., 2003, *Inside Macintosh: Speech Recognition Manager,* http://developer.apple.com/techpubs/macosx/Carbon/multimedia/SpeechRecognitionManager/Speech_Recognition_Manager/speech_recognition.pdf, Visisted April 25th, 2003.

[Standard 2002] Der Standard, 2002, *DVD-Player lädt Daten direkt aus dem Internet,* Der Standard, November 28th, 2002. See also http://derStandard.at/standard.asp?id=1145574.

[Standard 2003a] Der Standard, 2003, *Pläne für Umstieg auf Digital-TV werden konkreter,* Der Standard, January 15th, 2003. See also http://derStandard.at/standard.asp?id=1181802.

[Standard 2003b] Der Standard, 2003, http://images.derStandard.at/20030206/svidalinmoritz.jpg, Visited February 8th, 2003.

[STC 2002] Speech Technology Center, 2002, *Music Spotting – Real-Time Music Identification Technology,* ftp://ftp.speechpro.com/pub/leaflets/music_spotting.pdf, Visited April 25th, 2003.

[Tahara and Gaggioni 2002] Tahara, K., and Gaggioni, H., Sony Corporation, 2002, *MXF. Technology Enabler for IT-Based Broadcast Operations,* http://www.sonympeg-imx.com/resources/MXF_White_Paper.pdf, Visited March 3rd, 2003.

[Taiga 2003] Media Glue, *Taiga Content Delivery Solution,* http://www.mediaglue.co.jp/image/PDF/taiga-e.pdf, Visted April 10th, 2003.

[Tamura et al. 1978] Tamura, H., Mori, S., and Yamawaki, T., 1978, *Texture Features Corresponding to Visual Perception,* IEEE Transactions on Systems, Man, and Cybernetics (SMC), Volume 8, Number 6, pp. 460-473.

[Thompson 1999] Thompson, J., 1999, *This Is DAVIC,* DAVIC Information Seminar, Geneva, January 1999. See also http://www.davic.org/Info_Day/thompson.pdf.

[Tsui et al. 2000] Tsui, E., Garner, B. J., and Staab, S., 2000, *The Role of Artificial Intelligence in Knowledge Management,* Knowledge Based Systems, Volume 13, Number 5, 2000, pp. 235-239. See also http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/kbs-editorial-note2000.pdf.

[TV-Trawler 2001] Hunter, J., 2001, *TV-Trawler: DSTC Satellite Video Filtering Project Proposal (V1.0). Matching Video Channels to Personal User Preferences,* http://archive.dstc.edu.au/tvtrawler.html, Visited November 14th, 2002.

[UMTS-Forum 2003] UMTS Forum, 2003, *What is UMTS?,* http://www.umts-forum.org/servlet/dycon/ztumts/umts/Live/en/umts/What+is+UMTS_index, Visited May 5th, 2003.

[Utz 2000] Utz, P., 2000, *Digital VCR Formats Compared: Tabular Specifications,* http://videoexpert.home.att.net/artic3/256atab.htm, Visited March 3rd, 2003.

[Välimäki and Pitkänen 2001] Välimäki, M., and Pitkänen, O., 2001, *Digital Rights Management on Open and Semi-Open Networks,* Proceedings of the Second IEEE Workshop on Internet Applications (WIAPP 2001), 2001, pp. 154-155.

[Vollrath 2003] Vollrath, L., 2003, *Vorlesungsskript Histologie und Zellbiologie für Mediziner,* http://www.uni-mainz.de/FB/Medizin/Anatomie/Histologie/histokursws/HistologieZellbiologieWS200203Vorlesungsskript.pdf, Visited April 15th, 2003.

[VRML97 2002] International Organisation for Standardisation, 2002, *VRML97 – ISO/IEC 14772:1997, The Virtual Reality Modeling Language,* http://www.web3d.org/technicalinfo/specifications/ISO_IEC_14772-All/index.html, Visited April 24th, 2003.

[Wagner 2003] Wagner, R., 2003, *eLearning in der Medizin – Kooperation der bmbf-Projekte,* http://www.med.uni-giessen.de/agma/nmb-med/verbund.html, Visited April 12th, 2003.

[Wagner and Hansen 2002] Wagner, R., and Hansen, J., 2002, *E-Learning-Plattform für die Medizin. Bundesweise Kooperation,* Deutsches Ärzteblatt, Volume 99, Number 50, December 2002, pp. 3393-3394. See also http://www.deutsches-aerzteblatt.de/v4/archiv/pdf.asp?id=34879.

[Watson 1997] Watson, I., 1997, *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Morgan Kaufman Publishers, San Francisco, 1997.

[Weber and Schult 1998] Weber, G., Schult, T. J., 1998, *CBR for Tutoring and Help Systems.* In: Lenz, M., et al. (eds.), *Case-Based Reasoning Technology. From Foundations to Applications,* pp. 255-271, Springer-Verlag Berlin, 1998.

[Weber et al. 2001] Weber, R., Aha, D. W., Becerra-Fernandez, I., 2001, *Intelligent Lessons Learned Systems,* Expert Systems with Applications, Volume 20, Number 1, January 2001, pp 17-34.

[Werle 2000] Werle, P., 2000, *Active Documents to Support Work Processes in a Ubiquitous Computing Evironment,* http://www.dsv.su.se/research/k2lab/ucc/publications/huc2k.pdf, Visited September 14th, 2002.

[Werle and Jansson 2000] Werle, P., and Jansson, C. G., 2000, *Active Documents Supporting Teamwork in a Ubiquitous Computing Environment,* http://www.dsv.su.se/fuse/publications/PCC%20010403.pdf, Visited September 14th, 2002.

[Westbomke et al. 2002] Westbomke, J., Kussmaul, A., Raiber, A., et al., 2002, *Knowledge Management within Digital Libraries by Means of Personalization,* Proceedings of the I-Know '02 Conference, Graz, Austria, 2002. See also http://www.know-center.at/de/divisions/publications/pdf/slind-2002-IKNOW02.pdf.

[Williams 2002] Williams, J., 2002, *Practical Issues in Knowledge Management,* IT Professional, Volume 4, Issue 1, January-February 2002, pp. 35-39.

[Wipro 2001] Wipro Technologies, 2001, *Multimedia Home Platform. The Need for a Standard. White Paper.*

[Wollborn 2001] Wollborn, M., 2001, *MPEG-7 Systems Tools – An Overview,* 2nd MPEG Awareness Event, Singapore, March 2001. See also http://www.mpeg-industry.com/ifg/AEsingapore/aeSingPub/Systems_AE_Singapore.pdf.

[Woodward 2000] Woodward, P., 2000, *Introduction to MPEG-4 (Draft),* http://www.dcs.qmul.ac.uk/~petew/html/IntroMP4.html, Visited February 5th, 2003.

[WordNet 2002] Miller, G. A., et al., 2002, *WordNet: A Lexical Database for the English Language,* http://www.cogsci.princeton.edu/~wn/, Visited November 18th, 2002.

[Worrall et al. 2001] Worrall, S. T., Fabri, S. N., Sadka, A. H., and Kondoz, A. M., 2001, *Prioritisation of Data Partitioned MPEG-4 Video over Mobile Networks,* http://www.ee.surrey.ac.uk/Personal/S.Worrall/Publications/ett_pri_stream.pdf, Visited February 25th, 2003.

[Wu et al. 2002] Wu, S.-H., Day, M.-Y., et al., 2002, *FAQ-Centered Organizational Memories,* In: Dieng-Kuntz, R., and Matta, N. (eds.), *Knowledge Management and Organizational Memories,* pp. 103-112, Kluwer Academic Publishers, Boston 2002.

[XrML 2001] ContentGuard, 2001, *The Need for a Rights Language. Technical White Paper, Version 1.0.* See also http://www.xrml.org/reference/TheNeedForARightsLanguage.pdf.

[XrML 2002] ContentGuard, 2002, *XrML 2.0 Technical Overview. Version 1.0. March 8, 2002.* See also http://www.xrml.org/reference/XrMLTechnicalOverviewV1.pdf.

[Zacks 2002] Zacks, M., 2002, *Party's Over: Bills Come Due for Internet Radio,* IEEE Internet Computing, Volume 6, Issue 4, July-August 2002, pp. 12-13.

[Zhong and Chang 1998] Zhong, D., Chang, S.-F., 1998, *AMOS: An Active system for MPEG-4 Video Object Segmentation,* Proceedings of the 1998 International Conference of Image Processing, Chicago, Illinois, USA, 1998. See also http://www.ee.columbia.edu/~dzhong/Papers/icip98.ps.gz.

[Zimmermann 1997] Zimmermann, S., 1997, *Microsoft Agent and the #include Directive,* Extreme C++, Microsoft Interactive Developer, December 1997. See also http://www.microsoft.com/mind/1297/extreme1297.asp.
[Zloof 1975] Zloof, M. M., 1975, *Query by Example,* Proceedings of the AFIPS National Computer Conference, Montvale, New Jersey, May 1975, Volume 44, pp. 431-438. See also IBM Systems Journal, Volume 16, Number 4, December 1977, pp. 324-343.

Enhanced Index

The functionality of the index used in this thesis goes beyond a traditional, static index. It is enhanced in that it combines the content of this document with the power of the internet. If you are reading the electronic version of the thesis (PDF document) you can click on a keyword, and Google™, one of the currently most popular and most efficient search engines, is queried. This simple idea incorporates the internet as vast souce of knowledge into a simple document and facilitates the discovery of new knowledge.

Symbols

A

B

C

D

E

F

G

H

I

J

U

V

W

X