Centrum voor Wiskunde en Informatica

Centre for Mathematics and Computer Science

MULTIMEDIA

A ROUND TABLE DISCUSSION ABOUT THE STATE OF THE ART

Editors: H.P. Dijkhuis, B.P. Rouwhorst

CWI

MULTI MEDIA

MULTIMEDIA
The State of the Art
Round Table Discussion
11-12 April 1990
CWI, Amsterdam

Editors:
H.P. Dijkhuis,
B.P. Rouwhorst

CWI
P.O. Box 4079, 1009 AB, Amsterdam, The Netherlands

On April 11 to 12 1990 CWI organized a round table discussion about the state of the art regarding multimedia. This report summarizes the presentations given there, implying that the presentators have no responsibility regarding the correctness of the summaries.
Also a reproduction has been made of the plenary discussion, held on the last day of this round table meeting.

Key Words & Phrases : Multimedia, distributed systems, standardization, telematics.

Table of Contents

Introduction

by Paul ten Hagen[†]

The participants of this round table discussion have been invited by CWI because we want to start a major research effort in the area of multimedia systems. CWI is very pleased that so many people have responded by accepting the invitation. We have brought together active researchers from the multimedia frontier from well-known European research institutes, potential users and suppliers from Dutch industries and colleague researchers from the Netherlands.

The challenge for CWI concerning this research topic is, apart from the exciting prospects of successful multimedia facilicities, the multidisciplinary nature of multimedia research and the severe technological integration problems of multimedia which in themselves make multimedia a high tech subject. CWI has several strong research groups capable of contributing to multimedia research. The CWI R&D management now wants to build out of these a multidisciplinary team. This is no trivial task, especially not given the Dutch (research) culture of individual autonomy. We are convinced that the quality of research in computer science in the next decade will largely depend on the orchestration of research activities in order to produce interdisciplinary results. Moreover, only these results will stand a chance of being transformed into competetive products.

Current first generation multimedia has not been integrated well. This partly explains its limited penetration so far, although its potential has been recognized.
CWI hopes to establish through this meeting good working level contacts with colleague researchers and users. The size of the research field of multimedia is such that international collaboration is necessary for making sufficient progress with given limited resources.

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Audio/Video experiments at Xerox EuroParc

Summary of the presentation given by Mik Lamming†

Introduction

This presentation is about the research at Xerox EuroParc in Cambridge. This research institute exists already 3 years, of which 2 years in Cambridge. Its staff consists of about 30 researchers, ranging from psychologists and sociologists to technologists.

Charter, strategy and tactics

The charter of the research is to improve the effectiveness of information workers, in a broad sense.
The strategic goal to achieve this charter encompasses the support of inter-personal communications, the support of automatic capture and retrieval of problem-solving or teaching episodes, the encouragement of user involvement at all stages of the design cycle, a rapid prototyping and, last but not least, abundant ''What if...'' experiments.

From tactical considerations video is used to implement this strategy, because in the near future (video-) images will be digital, the current technological problems concerning video will be solved and the hardware will be cheap.
Video in itself has multimedia aspects, and it is inevitable that within a decade or so we will be surrounded by all kinds of communication tools. How can we anticipate that, and what are the requirements so that people can benefit from that technology?

The Video Project

To imitate such an environment we are trying to establish an electronic office at our laboratory, stuffed with video, and use that video as a communication channel.
To do so we first cover the building with video. Unfortunately there is in this moment no other solution than to use two networks, one for video and one for data. These two are difficult to combine. But this is seen as an interim solution. The audio/video has to be switch-controlled from a server on the Ether-net. A Sun workstation is used for that. Then every workstation and all public areas have to be provided with a monitor, a camera, a microphone and a speaker.

A first requirement then is that video communication resembles as much as possible human communication. An important aspect of this is that we try to maintain the coffee pot and hallway serendipity. The physical appearance of the system should fit in the buildings architecture. The system should be aware of the geography and, for instance, the timedifference between California headquarters and Cambridge. Then there is the issue of the connection costs (very important) and the protocol chosen to communicate. We try to make this as natural as possible.
Fundamental for human dialogues is the natural eye-contact; we had serious problems with looking at each other via the camera. This was because the camera didn't catch eye-contact when one was looking at the monitor to the person to whom he was speaking. This problem was solved by manufacturing a video-tunnel with camera and monitor in it and a transparent mirror, placed at some angle.
While communicating users can use little cameras and the mouse to point on the screen. By the way: the workstation screen can be very useful as a videodisplay.

The system should be non-invasive, that is, disagreeable interrupts should be avoided. And, of course, the system should be avoidable; if the user doesn't want to use the system, he should not have to.

Every workstation has a kind of *''communication-agent''* to make the connection to other workstations. You have to ''tell your agent'' with whom you want to communicate, and your agent arranges the contact with the agent of your opposite via the ''video-telephone'' (a piece of software). If all is authorized the switch-controller establishes the connection.

† Rank Xerox EuroPARC, 61 Regent Street, Cambridge CB2 1AB, U.K.

The next issue is the video storage. This includes all kind of security problems, for instance, who gets to see what? The quality of the video should be good enough to avoid questions like: ''What did he do/say?''. It seems difficult for a video system to focus on interesting things. Of course the costs are important factors concerning storage; one should calculate the costs for occupied space versus access-time/ability. The most difficult aspect of video storage is indexing; can this be done, and, if so, can it be automated?

It is obvious that user involvement in this video-project is indispensable. There have to be rational arguments for using video, for instance, one argument could be to get reasons why some decision were made. The feedback from the users is a welcome resource.

There was a problem how to get people used to it and how to create a system that can guarantee privacy and reply to the present standards of ethics.
The way this was solved is as follows: You start with creating a network by wiring the building. People are already used to wires through the building. Then you create video-walls in public areas and install workstations on the desks of the members of the projectgroup. Then take your time and show all interested people how nice, easy and efficient it can be to communicate this way. The effect can be that people become jealous and want to participate. With the (wanted) result that the project is pulled by people and not pushed to them.

When installed, people forget after some time that there is a camera. This makes ethics very important and if the problem of privacy is not solved then there will be no such system. Therefore, one of the main demands is that you are not allowed to look to somebody if the other is not looking at you.

A very fruitful way to do research is by means of ''What if...'' experiments. With these experiments the research group makes all kinds of assumptions and asks themselves questions. And, of course, tries to answer them.
Questions that arose from these experiments were, amongst others: Should there be video on every desk or wall? Why, when and how should we use video to communicate? This is a subtle, but very important question. How will the system change the way we work? Do we get an electronic architecture? Are we going to use it as video wallpaper, and use it for remote seminars?

Questions concerning video as data encompass the capture, storage, retrieval and processing of video-data.
Can it be used for education, analysis, video-mail, video-bulletin boards, or for making meeting minutes, diaries, notes?
How about automatic or semi-automatic indexing?

And last but not least we should realize that all this video might become ubiquitous.
How should we deal with sociology, for instance, with the reactions of people.
And where on earth should we keep these huge sets of video-data?

The Grif system

Summary of the presentation given by Vincent Quint†

Structured Documents

One of the research groups at INRIA is working on structured documents (already 7/8 years underway). The outcome of this research on structured documents can be of interest to multimedia.

International standards for the representation of structured documents, such as SMGL and ODA, don't fully understand the difficulties of structured documents (for instance, hypertext). One of the reasons for this is that they consider documents from a static viewpoint: they describe a document when it is exchanged between two systems, but they do not consider the processing of this document by a system. Neither do important concepts such as reusability and modularity appear clearly in these standards.

The research techniques used at our lab (experimental approach) is to work out prototypes for models of structured documents and to give it to users to get feedback.

The logical building blocks of a structured document consist of a user interface, lay-out description, software architecture and protocols for exchange.

The objects building up a document have different structures. They have to be integrated in multimedia. This may have severe consequences for the design of user interfaces. One can try to define a unique system to handle objects with different structures, but a problem, however, is how a user interfaces this.

Other problems in the dialogue between a user and the system are that users are only interested in a particular part of the system, and that there has to be interaction between the dialogue and the picture on the screen.
There is also the difficulty of integrating different types of data, which might lead to problems with the lay-out. It is difficult to describe a uniform way of representing these different datatypes. Today there is no language to describe the layout of such different datatypes.

A problem regarding the architecture is what editors are suited for these different datatypes? There is also the communication problem between editors and the databases containing these different datatypes.
An important requirement is that the system must be open-ended.

Also important is the impact of the model on the underlying pieces, user-interfaces, etc.

In the next section the Grif system is described. It is an interactive system for manipulating different types of documents and a first attempt to solve the problems mentioned above .

The Grif system

Grif is an interactive system for producing complex documents. It processes structured documents, which can themselves contain structured objects, for instance, tables, mathematical formulae and figures.

Documents and objects are considered as structures dynamically built following generic models.
These structures are used by the system for generating the image displayed on the screen.

Grif allows the user to define his own types of documents and objects and their visual appearance on the screen. There are possibilities to adapt the dialogue as the user prefers. In order to print documents existing formatters like T_EX, Scribe, Mint and Troff are used.

To allow real processing of documents, a high-level model is defined for Grif which essentially represents the logical organization of the document, not dependent on its physical structure.

Grif presents itself as an interactive system with possibilities for manipulating different types of documents like reports, books, articles, letters, forms, etc.
These are structured documents which contain chapters, section notes, headings with cross-references, tables of contents, indexes, etc.

† BULL-IMAG, Unité Mixte de Recherche, 2, rue de Vignate, Z.I. de Mayencin, 38610 Gières, France

They also contain structured objects like tables, mathematical expressions, drawings, flowcharts, programs, etc.
The system allows interactive working with a direct manipulation style of interface, not purely WYSIWYG. The system is not a collection of specialized editors, but an integrated system as well as open and flexible, extensible to various types of documents and objects.
The document model describes a representation of a document by a logical structure that contains elements like paragraphs, notes, sections, headings, chapters, references.
The specific logical structure of a document complies with the generic logical structure of its class. This generic logical structure defines the types of the logical components of a document and the relationships between these components. Some examples of classes are technical report, letter, thesis, etc.

A document can contain objects that are structured in the same way as a document: with a specific logical structure in conformance with a generic logical structure. Some examples of object classes are: Mathematical formulae, tables, programs, drawings, etc., but also paragraphs, bibliographic items, etc.

The presentation (graphical aspect) of a document is based on its logical structure. A presentation model associates presentation rules with each type of component defined in a generic logical structure. Several presentation models may be specific for each class, thus allowing the presentation of a document in different ways. A presentation model contains rules for presenting the logical attributes as well as possibilities for filtering in order to show only some parts of a document.
A presentation model specifies a numbering scheme for some types of documents (chapters, sections, notes, figures, formulae, etc.) as well as the way to display references (copy a part of the referenced component).

Editing.

The software architecture shows how to create an integrated environment with tools for writing, reading and handling documents with a high-level model.

Typical applications in the multimedia area are communication-, database- and editing-applications.
The editing application is the only connection between user and the file system. An editor has to communicate with the database system. Ordinary editors don't work with rules. Editing is just one of the instruments for applications on documents. Grif as an editor has parts that are useful for different applications. It is not one large program but an editing toolkit. In Grif pages can grow by editing. The logical structure is important and has nothing to do with pages.
On the first level there are a lot of functions like *load, save, open, view*, etc. of documents.
Working on the second level you can handle links between elements of structured documents, and functions of logical parts of a structure have been provided.

For multimedia documents you have to develop an editing toolkit first, then define applications.

Status.

At this moment there are only three types of contents: text, rastergraphics and structured graphics. Associated elements can be pictures and sound.
The presented document model can not proceed hypertext. Hypertext has more relations than shown in the document model, for instance, references to persons.
There are questions about the author point of view. It is obvious that it is difficult to edit this kind of structure organized in a tree.
The frame of a document is fixed, for instance, cannot be changed by the end user. That is good if you want to use a given model, for instance, with technical documentation. But the weakness of these types of structures is that you have to change everything, for instance, book to article, paragraph to section, etc. So it is difficult to transform the logical model. Also a difficult part is the generation of the graphical part of the structure.
Further possibilities in Grif are: the language P, multi-windowing system, same logical structure in different presentations, and the possibility to use 'dumb' terminals.

Research at BERKOM

Summary of the presentation given by Carsten Wieschiolek†

Overview

BERKOM (BERliner KOMmunikationssystem) is a German company run by DETECON, a daughter of Deutsche Bundespost. An overview of its research-activities is given here.

The objective of BERKOM is to stimulate the development of services, endsystems and applications for the future Integrated Broadband Telecommunication Network (ISDN-B/IBFN).
The problems encountered in this field are that there is no clear idea about applications, there is no international standard, there are no units or endsystems available, and the technology is not powerful enough.

The development of the communication network includes satellites, telephone nets, and integration of videotext and -images.

BERKOM takes the following steps to achieve its goals:

- *Marketing analyses*, tools for demand evaluation and analysis of future broadband potentials. Applications of and demands on new multimedia office communication systems. Main study analysing application and market potentials.
- Realization of *applicable demonstration projects* to be used as soon as possible in the forerunner network. Examples of such projects are:
  - *Telemedicine* (as part of the EG-project RACE), including requirement analysis, dedicated workstations, multimedia endsystems, MEDIKON and RADKOM.
    The aim is to be able to interactively query, at great distances, large medical databases, containing data about patients, scanned images, computer tomographical data, etc.
  - *Telepublishing* (also as part of the EG-project RACE), including requirement analysis, printing on demand, for instance, mailorder catalogue (with 50 Tbytes content). It is sometimes better to transport electronically than have hardcopy printed catalogue distributed.
  - *Distributed Factories*
  - *Broadband Information Systems*, including the Broadband-Information-System ''Berlin-Info'' (BIS) and AKUBIS.
  - *Office Systems*. Central storage and retrieval systems.
- *Development of Services and Standardization.*
  There are scientific projects, like BERMMD and BERMAN, and also working groups that elaborate proposals for the international standardization of telematic services (*BERKOM - Reference Model* for integrated services).
  There is an active participation in ISO.
  In the EG-project RACE, BERKOM is working on the EBIT (European Broadband Interconnection Transfer) Service Project. Any data can be transported. Transfer speed about 2Mbits/sec.
- *End systems*. There are studies on Man-Machine Interface, Display Architecture and Market Analysis in this field.
  The work on multimedia end sytems includes the development of a multimedia workstation, video-codecs (2 Mb/s and 140 MB/s), and a color-scanner and -printer.
  For this project BERKOM works on multimedia datastructures (for users), including text, geometrical graphics, raster images, audio/voice, video/film, and on communication services. Each service can have its specific (abstract) datatype.
  BERKOM is working on a prototype for the Forerunner Broadband Network (FBN) operating test of the DBP, and is developing its FBN-endsystem.

† Detecon, Technical Centre Berlin, Voltastraße 5, Haus 10, Aufg. 2, 1000 Berlin 65, B.R.D.

- *Networks.*
  There are studies on broadband switching and transmission principles, ATD and Virtual Circuit Switching (VCS).
  There is a Test Network under development, including ISDN-B Switches, an ATM-Switch, TA's with R-Reference Point and a connection with FBN. It uses two distinct transport systems, the standardized transport-system (ISO-OSI), and a test net with STM/ATM switches. (STM: Synchronous Transfer Mode, ATM: Asynchronous Transfer Mode).
  Finally, work is done on the Gateways Endsystem Connection, including ISDN-B Network adaptor boards, protocol adaptor boards, ATM adaptor boards, FDDI-1 adaptor boards, and the project BERGATE.

Discussion: Is it really cheaper to transport Tera-bytes of information through an advanced broadband network? Communication costs of high quality multimedia can be very high. Sending Tbytes of information should be done by trucks, use the broadband for *interaction*, that's important.

In the following sections two pilot projects of BERKOM are presented in more detail, namely the BERKOM Test Network and the BERKOM Reference Model.

BERKOM Test Network

We are dealing with different datatypes and different networks.

The aim of BERKOM is to design and develop a B-ISDN global network. In this context a global network implies a network architecture not only for layer 104 in the ISO/OSI Reference Model, but also new applications, techniques and multi-functional endsystems in this new environment. Layers 1-4 deal with the management of connections and the transport of data.

The BERKOM test-network consists of fibre-optic cables of approximately 8000 km length, connecting the major research- and development institutes and computer manufacturers. The BERKOM switching nodes provide the subscriber channels for 64 Kbits/s, 2 Mbits/s and 140 Mbits/s, conforming to the PCM hierarchy and the H_X-channel structure /CEP 86/. The D-channel protocol, as defined for the the ISDN by CCITT, can be easily enhanced to support the establishment of the circuit switched channels with the above mentioned characteristics.

The channel structure provided on a subscriber line will be $S_B=1*D+2*B+4*H_1+1*H_4$ where

D = 16 Kbits/s

B = 64 Kbits/s

H_1 = 1920 Kbits/s

H_4 = 135.168 Mbits/s

at a total transmission speed of 153.6 Mbit/s.

The switching nodes for the BERKOM test-network are provided by Standard Elektonik Lorenz and by Philips Kommunikations Industries. The ATD pilot switch is provided by Siemens AG. As the standardization of the H_4-channels has not yet been finalized and discussion concerning line speeds and channel structures is still in progress, a terminal adaptor with an R-interface point for each end-system shall provide end-system decoupling. The switches nodes are pilot developments and each will serve up to 3 end-systems.

Problems include how to synchronize the channels, how to combine different media-parts and how to present them. Datatypes and channels have to be synchronized.

Switching technology is also a problem. How many Mb one needs in a network depends on datacompression. With CODEX you need about 35 Mb.

BERKOM Reference Model

The reason for developing the Reference Model is that a pragmatic solution, permitting the realization of distributed applications based upon standards and application profiles, must include the selection of suitable combinations of standards and application profiles. Consider it as a ''How to use Standards''-guide.

The main purpose of the Reference Model is to identify common functionality among distributed applications, in order to combine a number of components to serve different application areas.

Combinations of components are used to create profiles that can be subdivided into

- data structure profiles,
- communication profiles, and
- profiles for components in the application environment.

The communication profiles must be suitably combined with the transport system profiles as defined in the Reference Model for lower layers.

The profile recommendations are based on the OSI Basic Reference Model, existing CCITT telematic services, distributed applications already standardized within OSI, and other international standards and profiles in the areas of communications and data structures for information interchange.

Hereby, existing telematic services are taken into special consideration. The telematic services cover complete service descriptions, including communication protocols, interchange formats and terminal characteristics. The present version of the Reference Model II concentrates, however, on communication aspects and interchange formats.
In addition to these profile recommendations, areas are identified for which the available standards do not offer any solutions yet.

An important characteristic of information interchange is the information types supported. Typically, character text, geometric graphics and rastergraphics (images) in various degrees are mandated by today's document processing systems. Developments in recent years in the area of workstation- and network- technology have made it possible to include time-variant information, such as moving pictures and audio. As these five information types are used to present information in many application areas, they are considered to be application independent and are called basic information types. Additionally, many applications require the interchange of specific information that must undergo further processing before being representable as basic information types (for instance, product definition data), or that is not intended to be human perceptible at all (for instance, robot control). Accordingly, the notion of application-specific information types is introduced.

The term multimedia is generally associated with information in a form acceptable for humans, i.e. representable through basic information types. However, as many applications require the integrated processing of basic and application-specific information types, multimedia information in the Reference Model is understood in a more general sense to include application-specific information types.

multimedia information comprises basic and application-specific information types, i.e. it is intended for visual and auditory perception as well as for automated processing. The information is not restricted to local storage, but may be distributed. The information may, but does not have to, be presented on different presentation media.

In addition to specific information types, general information structuring capabilities are of importance.

In addition to recommendations in connection with the basic information types, presentation-independent and presentation-specific structuring capabilities are indicated as well in ODA. For example, logical and layout structures in ODA.
This structuring capability, in its ability to express the interplay of more than one (not necessarily different) basic information types, will be of most interest when selecting appropriate standards or profiles for multimedia applications.

An application-specific information type has to be transformed into basic information types to become human perceptible. This is because the intention of product information is, in the first place, to describe the technical product for special purposes like manufacturing, simulation, etc., and not to derive a representation suitable for human perception. If the latter case is required, it can be done by preprocessing, but in general a certain amount of product information is lost.

In addition to communication-specific aspects and data-structure aspects, telecommunication-specific characteristics, such as terminal characteristics of telematic services, that are out of scope of OSI, are in the scope of this Reference Model. This leads to the need for standardization of interrelationships between different parts of an application. The following classification is given:

- communication specific profiles
  - OSI application profiles
  - advanced application profiles
    encompasses light weight protocols for isochronous communication and multipoint communication, such as multi-casting.
- application environment profiles
  encompasses non-communication specific aspects such as filing and retrieval security, presentation of multimedia information and terminal characteristics.

The combination of data structure standards and profiles with the communication and application environments standards and profiles build the basis of any distributed application in the scope of this Reference Model.

The profiles which are derived for, and should be used within applications, are called application profiles. The applications can be constructed from these parts given by the Reference Model:

- data structure profile (interchange formats)
- communication profiles and
- profiles for components in the application environment such as terminal characteristics,

Characteristics for OSI application profiles is that the communication protocols are defined without extension of the capabilities and definitions of the OSI Basic Reference Model. Advanced application profiles encompass:

- real-time communication support
- multi-point communication support, such as multi-casting.

Real-time communication profiles are used for real-time audio communication and real-time moving picture communication. From the service point of view the service is an OSI extension, and from the protocol point of view the protocols are reduced in their complexity.

The combination of components described by profiles is necessary because some applications require the functionality of more than one application profile. The combination of profiles requires an adequate modelling approach. Solutions for the following working areas must be developed:

- The main task from the communication point of view is the development of solutions for the coordination of isochronous and anisochronous communication, including the synchronization of isochronous communication, such as audio together with moving pictures, if no integrated code is used.
- Also from the communication point of view, more than one application service element in one application process have to be modeled.
- If profiles of the environment are required from applications and have to be combined with communication profiles, further modelling approaches are needed:
  - In some cases interworking between profiles must be defined.
  - The DOAM (Distributed Office Application Model) activities can be used in the office environment.
  - More general approaches will be developed in the approach based on ODP (Open Distributed Processing) and DAF (Distributed Applications Framework).

Requirements not fulfilled or under development by existing standards and profiles are listed below, not necessarily exhaustive:

- Data structure area
  1. moving picture,
  2. integration of moving pictures and/or audio with the other basic information types,

3. integration of application-oriented information types,
4. advanced functionality of the basic information types,
5. advanced document architecture regarding synchronization, alternative structures, etc.
6. external references,
7. variants,
8. versions (revision history),
9. tables,
10. forms,
11. colour reference model,
12. role support in group communications.

Standards under development in this area are:

— ODA extensions,

— DATAM, CCITT SG VIII,

— HDTV [HDTV1],

— Image Interchange Format (IIF),

— Digital Audio and Picture Architecture (DAPA) of ISO/IEC JTC1/SC2/WG8 and CCITT New Image Communication (NIC) Group working on Raster Image Communication Architecture (RICA),

- Communication area

1. development of advanced profiles for moving pictures and audio,
2. synchronization mechanisms between isochronous communications (for instance, moving pictures and audio),
3. synchronization mechanisms between isochronous and anisochronous communication,
4. solutions for combination of different communication profiles.

Work under development in this area is:

— Distributed Application Framework (DAF) [CCITT-DAF],

— Digital Audio and Picture Architecture (DAPA) of ISO/IEC JTC1/SC2/WG8 and CCITT New Image Communication (NIC) Group working on Raster Image Communication Architecture (RICA),

— solutions as part of the development of this Reference Model.

Discussion

Standardization and research should not go in parallel. But stimulated by industry it is often shown that research has been influenced by the call for standardization. The list of existing standards that won't work is long. These days one cannot research with EG-funding without a standard and that is not the right way. The industry has to say when to define a standard; that work must not be done by researchers. Experience in the UK has shown that waiting for standards has delayed research.

The MULTOS Project

Summary of the presentation given by Martin Kersten†

Overview of MULTOS

History

MULTOS was included in the former Esprit 2 TROPICS project (2 years ago), in which CWI was involved.

Objectives

It started as an Esprit 1 project with the following objectives:

- Management of multimedia (Text, images, voice): Compound Documents.
- Definition and use of a Document Model for efficient Document Storage and Retrieval.
- Definition and use of content-based retrieval techniques for text and image data, without knowing exactly the structure.
  MULTOS focused on retrieval of data based on Imaging Techniques.
- Proposed use of standards for document exchange (ODA/ODIF).
- Use optical discs for storage of large amounts of data.
- Use of techniques for Automated Document Classification.
  Eg. if the string ''*tel.:*'' is encountered one can assume that the digits that follow will make up a telephone number.

Summarizing: MULTOS is focused on document filing and retrieval using a novel document model and system architecture.

Discussion.

Documents are scanned and classified through annotation and partial pattern recognition. There is a prototype implementation. Aim is a marketable product.
Connection with multimedia: How to connect text and images, how to make (for instance) a GKS-representation out of it.

The office documents can be classified in private ↔ public and updatable ↔ stable.

The MULTOS global software architecture is based on the Adrew toolkit.
The keys (search) can be adjusted afterwards.

The keypoint in the retrieval capability of the system is that some queries are not exact but approximate in nature.

Where is the multimedia aspect? There was no intention to start from multimedia.
Are there fundamental differences in their approach between images and text, or did they just start with text because it is easier. The impression is the latter. Some multimedia aspects can first better be applied/tried on text and later be expanded to images with their typical difficulties.

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Detailed Description of MULTOS

This description is a summary of the article *MULTOS — a Filing Server for Multimedia Documents*, by E. Bertino, F. Rabitti, C. Thanos, A. Converti, P. Savino, H. Eirund and K. Kreplin. This article was published in *Information Technology for Organisational Systems*, H.J. Bullinger et al. (Editors), Elseviers Publishers B.V. (North Holland) © ECSC-EAEC-EEC, Brussels-Luxembourg, 1988.

Introduction

The computing power of the 90's will permit user applications that require processing and distribution of information in several media, including alphanumeric data, text, voice, computer graphics and images, that is the development of computer-based multimedia information systems. Important components of an office computer-based multimedia information system are: multimedia editors, multimedia information communications, multimedia information filing and retrieval. Work toward the development of multimedia editors and multimedia information communications is well under way and standards for the interconnection of the multimedia systems of the future are already evolving. On the contrary, the work toward the development of a multimedia storage and retrieval tool is at the very beginning, in spite of the increasing demand for filing and retrieval of large amount of multimedia documents coming from the office applications. There are some difficult technical problems that should be solved to achieve significant progress toward the building of a multimedia storage and retrieval tool. For instance new office models are needed to overcome the integration problem derived from the fact that several different data types should be handled in a uniform way. To provide content addressability, since access based on a unique identifier is clearly inadequate, some important issues should be addressed. For instance content addressability in text may have to deal with spelling errors. Content addressability in images may have to use indirect techniques, similarity retrieval techniques, adaptive techniques or spatial relationships. Automatic information extraction techniques may be needed. Content addressability in voice data is still bound by technology limitations. Operational issues like concurrency control, recovery, security, version control, and storage techniques for large, multimedia documents have also to be investigated.

The MULTOS project aims at the development of an advanced multimedia filing and retrieval system and therefore addresses many of the above issues. In particular the issue of system architecture, document representation, content addressability, storage organization and query processing are addressed. In MULTOS document retrieval is based on a document model which introduces a concept of document type defining the document layout, logical and conceptual structures. The document layout and logical structures are kept within the ODA standard. The conceptual structure describes the user semantic view of the document with respect to document retrieval by content on a high level of abstraction. With the MULTOS document model there is the problem of creation of the conceptual structure. Although it can be constructed step by step by an editor during the creation of the document, this possibility is not sufficient. An open system must be capable of accepting documents from arbitrary sources, including documents without a conceptual structure. Therefore the MULTOS system provides a classification subsystem that automatically constructs a conceptual structure based on a given set of type definitions.

Document Model

In the context of MULTOS a model for the representation of multimedia documents and their semantics has been developed. To support different operations (i.e. editing, presentation, retrieval) the document model supports several structural descriptions of a multimedia document. The logical structure determines how logical components, such as sections and paragraphs, are related. The layout structure describes how documents components are arranged on an output device at presentation time. There may be couplings between logical and layout structures. The document model adopts a standardized representation based on ODA (the Office Document Architecture, a standard under definition by ISO) for the logical and layout structures. The conceptual structure in the MULTOS document model is defined by a formalism called Conceptual Structure Definition (CSD). Documents can be grouped into types by means of their Conceptual Structure Definitions. A type describes only the common conceptual components of the documents of this type (i.e. a type is again a CSD). The CSD type concept allows several levels of refinement when defining types. A new type is refined by refining (specializing) an existing type and thus becomes a subtype of this type. So there is a relation of generalization defined on types, that structures the type catalogue into a hierarchy. This relation is called the ''is_a-relation''.

MULTOS server architecture

The MULTOS document server is based on a number of autonomous Clients and Servers. Two document servers are considered to provide different filing capabilities.

- *Current Server.*
  In the office environment exist documents shared between several users and that require frequent modification. The MULTOS current server allows the insertion, deletion and modification of public documents. Moreover the system supports queries by content with a fast response time.
- *Archive Server.*
  It allows to deal with stable and infrequently accessed documents. The amount of documents with these characteristics is usually very large. The server takes advantage of the use of Optical Disk media to store this large amount of data. It supports queries by content with acceptable response times. The internal structure of a document server consists of the following components:
- *Dialogue Manager.*
  The Dialogue Manager is responsible for all server/client communications.
- *Server Controller.*
  The server controller coordinates all the activities inside the server.
- *Type Handler.*
  The type handler maintains document type definitions and manages all the operations on types.
- *Collection Handler.*
  This module handles collection definitions and manages all the operations on collections.
- *Classification Handler.*
  The classification handler analyses incoming documents received from foreign clients such as gateways to external systems in order to automatically associate a type and collection.
- *Query Processor.*
  The query processor is responsible for the execution of queries.
- *Structure Translator.*
  The structure translator handles the conversion between the ODIF representation of documents (exchange format of ODA documents) and the storage format in the Document Storage Subsystem.
- *Storage Subsystem.*
  It provides functions for the storage and retrieval of ODIF document objects, maintains access structures based on the CSD for the retrieval of documents by attribute, and allows document retrieval by content.

Content based retrieval

In MULTOS content based retrieval is obtained by allowing to specify in queries constraints on the document type, its collection, values associated with a conceptual component and restriction on its content. The textual part of a multimedia document is the prime candidate for the specification of content restriction. However, techniques for the use of the image part of the document are currently under investigation.

Query Language and Query Processing

The query language defined for the Multos is based on the document conceptual model and on the content based retrieval techniques. Queries may have conditions on both the text content and the conceptual structure of the document. In particular expressing conditions on the CSD means requiring documents having the conceptual component whose name is specified in the condition.

The processing of a query goes through several steps. After the initial parsing, some query transformations are performed based on the type hierarchy.

Automatic Construction of Documents' Conceptual Structures

An essential quality point of the system is its document classification, i.e. the ''proper'' construction of conceptual structures of documents. With this structure a document is allocated to the type that describes the semantics (i.e. the structure) of the document most precisely. This task requires document analysis facilities beyond the syntactic level, but because of the extreme complexity of this (unsolved) problem must fall short of total 'text understanding'. In MULTOS a middle course is adopted. With the use of a priori defined conceptual types that contain document content constraints, documents are analyzed, structured and classified.

The Classification Approach

The basis of the proposed classification process is a predefined type, whose conceptual components the system tries to recognize in the document. By the instantiation of the type CSD we get an appropriate document CSD with the specialization rules applied to and all basic components liked to the relevant content parts. To find the best fitting type, the type hierarchy is investigated top-down. For each type the system tries to recognize if the document includes the specified components. If this is the case, all subtypes of this type are investigated. With this procedure we get the best fitting type for a document, together with a most specialized document CSD that can be constructed for this document as an instantiation of this type.

Architecture of the Classification System

The classification system, composed of the classification handler and the knowledge bases, gets the type definitions and a document as inputs and produces the document's conceptual structure as an instantiation of one of the types. The classification handler itself consists of a classification control unit and a document analysis component. The classification control unit traverses through the type hierarchy, controls the construction of the document conceptual structure and calls the document analysis component to satisfy the CDL predicates, i.e. to interpret the CDL program code. The knowledge the system operates on can be subdivided into: Structural knowledge: the knowledge about dependences and constraints regarding data. Application Specific Knowledge: the expert knowledge about the application environment of the system. General Knowledge: the knowledge independent of the application environment. Actual data: the document to be classified itself.

The TROPICS Project

The following is a summary of the article *The Design of TROPICS-SQL for Office and Cartographic Applications*, by M.L. Kersten and M.H. van der Voort[†]. The work reported in that paper was conducted as part of the TROPICS project, i.e. ESPRIT II project 2427.

Introduction

Despite the success of the relational model in the commercial environment, many limitations of the relational model have been uncovered over the last decade when applied to non-traditional information systems, such as Office Automation, cartography, CAD/CAM, etc.. The current awareness is that an enhanced datamodel and a more flexible architecture is needed to cope with the problems effectively.

One of the first serious attempts in this direction is the Non-First Normal-Form (NF2) model proposed by Schek and Pistor. They observed that in many applications there is a need for set-valued and list-valued attributes. Consequently, they dropped the requirement for a relational scheme that each attribute is atomic, i.e. non-decomposable. The result is a datamodel for hierarchical structured objects with a declarative query language.

This approach has been generalized into Extended NF2, where all restrictions placed on the type constructors have been dropped. Moreover, an SQL-type of query language is defined.

The main limitation of the NF-2 approach is that it focuses primarily on the structuring aspects, while ignoring the behavioral aspects of complex objects. These behavioral aspects form the focus of object-oriented databases with the salient features of abstract data types, object identity, type inheritance, and persistency independence. Abstract datatypes enable the encapsulation of data and their methods, while hiding their implementation details. Object identity associates an unique identifier with each object stored. This identifier can be used to model object sharing and thus to build graph-like objects. Type inheritance enables objects of different types to share part of their behaviour, i.e. their methods. This encourages re-use of design and code. Finally, persistency supports the survival of an object in the face of system malfunctioning.

The area of object-oriented database is strongly influenced by object-oriented programming concepts. Therefore, many analogies can be drawn between programming concepts and database concepts. Moreover, some researchers have taken the route to enhance an existing object-oriented programming environment to arrive at an object-oriented DBMS. The prime issues to address are support for persistency, concurrency control, recovery, and declarative query formulations.

The main stream of research invested in post-relational systems tries to support specific domains. In our case, we are constrained to support an office environment and cartography applications. The office application deals with large collections of semi-structured objects, that stresses the capabilities needed for data definition and data manipulation. For example, a *union* type constructor is needed to model tables that contain both letters and memos. Moreover, in querying a document base one does not necessarily access the database by collection name and descending its structures, but rather by presenting a set of <property, value> pairs. For example, find all the documents in the database that refer to the product MilkSoap. Such a query stresses the functionality of a query processor, because it should first locate all objects that deal with products and, thereafter, select those that satisfy the query condition. Alternatively, access methods based on text pattern recognition, normally found in information retrieval systems, should be used to find the data of interest.

The cartography application aims at improving the production of thematic maps, a data- and cpu-intensive application, using database technology. Furthermore, supporting a geographic application stresses the architecture and implementation of the database system. For example, one has to extend SQL towards a more object-oriented language, such as to introduce the new concepts (classes, inheritance, methods, and message propagation). Combined with an extensible DBMS framework, it then becomes possible to satisfy the high processing demands of such an application.

---

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Requirements for TROPICS-SQL

We introduce the three application domains dealt with in the TROPICS project. Each application domain is studied for the requirements it imposes on the database language. The presentation of the domain is necessarily short and incomplete, because we are primarily interested in the database concepts.

Office Document Modelling

The first application domain that impacts the definition of TROPICS-SQL is the storage and retrieval of documents in an office environment. This domain has also been the target of the MULTOS project. It has produced a prototype system for the efficient storage and retrieval of such documents. Its characteristics make it suitable for large applications, such as in the Public Administration area, banks, etc., and for small environments.

Much of the document processing inside MULTOS is based on a document model supporting semantic oriented descriptions of documents. This semantic document representation is the basis for the MULTOS content oriented document processing and retrieval. The project objectives are characteristic for this application domain and contain, amongst others:

- the definition of a multimedia document model to be used for document filing and retrieval.
- to investigate and to implement knowledge-based techniques for automatic document classification.
- to provide document management functions like access control, security, integrity and version management.
- to implement techniques for content based retrieval on text and attribute data and investigate content based retrieval.
- to develop an efficient and cost effective system for filing and retrieval of multimedia documents.

Office documents including different kinds of primitive data types, also called multimedia documents, are generally structured as a hierarchy of related document components. Moreover, the resulting objects are more complex than usually managed in a 4GL form manipulation environment. In particular, they do not fit the ''strong'' type concept being used. Instead, documents tend to differ widely in structure, making a relational system cumbersome to use as a direct storage medium.

Requirements for TROPICS-SQL

The MULTOS project was largely application driven. In particular, the document model and system features are biased towards a specific implementation. However, many ideas can be generalized to accommodate a wider class of applications.
Moreover, the MULTOS interface between Client and Server can be loosened by introducing TROPICS-SQL. The prime benefit for the MULTOS project would be a more open system. Then, the documents can be easily related to other stored business data, such as the data stored in a relation table that originates from a decision support system.

The prime requirements derived from the MULTOS project that should be taken into account for the design of TROPICS-SQL are :

- *Basic Types*
  It should extend the set of build-in types with those to maintain and analyse document images.
- *Hierarchical Objects with Union Types*
  It should support the description of complex hierarchical objects.
- *Object classes*
  It should support the notion of collections of objects, possibly of different type.
- *Object versions*
  It should support a versioning method covering the history of the document as a whole, but also its individual parts.
- *Integrity Rules and Flexible Integrity Maintenance*
  It should include a powerful mechanism to describe and to enforce integrity rules.

- *Querying by Weak-Types*
  It should include a powerful view mechanism to isolate all objects of a particular 'weak type'.
- *Functional Extensions*
  It should include a mechanism to introduce new operators and functions, which can be implemented either in terms of the query language or with an external implementation language.

Geographical Information Systems

The focus of a geographical information system (GIS) developed in the context of our project is to enable the efficient management of spatially-referenced data of various kinds and different origins. The information managed within a GIS is conventionally divided into two categories: alphanumeric encoding of semantic data and spatial encoding of objects. The former mostly describes administrative data, such as the name of a road, number of inhabitants, and area use. The latter describes the corresponding spatial object in terms of its absolute/relative position within a 2(3)-dimensional space.

Requirements for TROPICS-SQL

The requirements posed by this application can be generally phrased under the heading ''extensible database system'', because it stresses the implementation of a relational system more than it stresses the SQL interface. Following, the prime requirements derived for TROPICS-SQL are

- *Abstract Data Types*
  it should support the definition of abstract data types which are either implemented in terms of multiple SQL 9tatements or within a conventional host language.
- *Operator Overloading*
  it should support operator overloading to re-use and capitalize on their common interpretation.
- *Complex Objects*
  it should support the notion of complex objects. In particular, it should be possible to describe directed graphs.
- *Performance*
  it should provide high-performance database operations.

General Objectives

The objectives derived from experiencing the application domains has led to the following overall design objectives for TROPICS-SQL. A prime concern is to arrive at a reasonably balanced solution, that leaves room for specialization to accommodate individual domains.

SQL Compliance A prime concern is to be backward compatible with SQL.

Object Constructors The type constructors provide in TROPICS-SQL should cover the cases described in the preceding sections. Versions are excluded for the time being, since a general semantics is not yet fully understood. Similarly, the inclusion of a trigger mechanism to handle integrity is postponed.

Object Type Selections To accommodate concise queries, we should support a query mechanism on the database scheme as introduced for the document model.

Extensibility The extensibility required for non-trivial applications requires a mechanism to (interactively) specify methods within the SQL framework. The implementation of methods range from multi-statement queries with flow-control, to externally implemented routines. Furthermore, a strict adherence should be made to the separation of external interface and the implementation of an abstract data type.

Object Server Platform The platform on which TROPICS-SQL is to run should be of no concern for the language definition.

Overview of the Data Model

As previously mentioned, TROPICS-SQL is a derivative of the SQL standard. Thus, we take knowledge on its syntax and semantics for granted and we will only emphasize the deviations.

Basic Types

The SQL standard has a limited set of built-in types that accommodate the business applications. Many commercial systems have extended this standard list, so as to include currency and date domains. Sometimes augmented with a plethora of display and arithmetic operators.

The approach taken in TROPICS-SQL is to keep the basic set as small as possible and to rely on supporting the more complex cases through the Abstract Data Type facility.

Type Constructors

The main deviances from SQL are visible in the way the user can define complex objects in TROPICS-SQL. In addition to the tuple constructor of SQL, it provides for the construction of arrays, lists, bags, sets, and reference types. Moreover, the type constructors can be used recursively to describe all data structures encountered in programming languages.

A significant departure from existing object-oriented database systems is the support of union types. They are required by the office applications.

Literal Complex Objects

To improve the expressiveness of the language, a mechanism is needed to construct specific objects out of the literal representation of their underlying types. In TROPICS-SQL this is supported in full. Tuple values are merely constructed as in SQL by parenthesizing a list of values. The user is not obliged to fully specify the type of the object components. In many situations, such as within an update statement, the compiler can infer this information from the context and the type catalogue. Yet, the type names may be added to improve visibility of the type structure as well as to circumvent ambiguous interpretations.

Creation and Manipulation of Global Objects

The SQL create table statement has been generalized to define a table as a bag of objects described by a complex object type. Objects can be inserted into (deleted from) this bag and SQL-like queries can be phrased retrieving part of this table.

The global objects dealt with in an SQL environment are the tables. In particular, the only way to define a variable that holds a single value is by defining a unary table with a single row. In TSQL this restriction has been dropped. The user can introduce any kind of global object through the create statement. The table introduced in the previous section are just special cases of global objects.

Each object inserted into the database is assigned an unique object identifier, called the **ref** of the object. An object identifier is also attached to every (sub) object in the database. Following, an object identifier can be used as a reference to another (sub)object in the system. The object identifier is comparable with a safe pointer in programming languages.

The focus on tables in SQL has resulted in a limited set of manipulation statements. One can insert tuples into and delete objects from a table only. Since TSQL supports other types beside tables, a more general insert and delete concept is supplied. The equivalent of the SQL insert statement is a *set* statement, which is similar to the assignment statement in programming languages.

Querying Facilities

The TSQL query language facilities have been extended in several directions. First, the well-known SELECT-FROM-WHERE construct has been extended to deal with complex objects. A novel aspect provided for is to search objects in the database using a restriction on their type structure.

The syntax and semantics of the SQL data manipulations have been left intact for the SQL subset of TSQL. That is, querying tables, i.e. bags of flat tuples, can be written with the old-fashioned syntax and produces the expected SQL result.

The components of a complex object, i.e. its attributes, can be accessed with a path notation. To avoid ambiguous interpretations we require that the components of a (local) type have unique names.

Type Selections

We introduce a mechanism to limit the search space for database queries using a restriction on the types associated with the objects.

Selections are focussed on analysis of the value associated with an object. The exception being the **case**-construct that first analyses the type structure to determine the required value. This type analysis can also be applied within the **from** clause to limit the objects for evaluation to those that satisfy a type restriction. Like a 'weak' type in the MULTOS document model, such a type restriction is a boolean predicate that describes the minimal structure of the objects to be considered.

Another powerful type selector in TROPICS-SQL is to zoom-in on object components anywhere in the database that are recognized as belonging to a specific type.

One of the problems encountered in using SQL-like languages proposed for NF2 is that part of the query condition has to be repeated in the **select**-clause. The underlying cause is that in SQL there is only one place to define a variable, namely, within the **from** clause. Moreover, one can import a tuple binding into an inner query, but one cannot export a binding from an inner query block.

The view statement in TSQL is a generalization of the corresponding SQL construct. In particular, the query part may refer to the name assigned to the view. This way, the view can be interpreted as a linear recursive query.

Abstract Data Types

The prime requirement posed by the geographical application is to allow for the definition of new datatypes with its operators. This is covered by the Abstract Data Type specification of TSQL.

An Abstract Data Type can be viewed as an extended type description. On the one hand it describes the external interface of the abstract objects in terms of visible structure and applicable operators. On the other hand it describes how this new data type is being implemented.

Currently the implementation of each method is either a multi-statement TSQL query or a reference to a routine written in an external implementation language, such as C/C++.
A method can be applied to an object if the TSQL compiler can infer a unique path through the object structure to locate the method definition.

Amoeba – High-Performance Distributed Computing

Summary of the presentation given by Sape Mullender†

Introduction

An Amoeba typically consists of four kinds of machines: personal workstations, dedicated server machines, a processor pool, and gateways that connect to other Amoebas via wide-area networks.

Amoeba is a capability-based distributed operating system. Each object has a capability which identifies the service that manages the object, and the object itself within the service. Presenting the capability for an object to its service is taken by that service as *prima facie* evidence of the client's right to carry out operations on the object.

Amoeba machines run a copy of the Amoeba Kernel, a small operating system kernel providing process management services and interprocess communication facilities. Other services, traditionally in the operating system kernel, such as the file system, are implemented as user processes.

Research on the Amoeba system focuses on several essential properties of distributed systems: security, availability, fault tolerance, scalability, and the possibility of widespread geographic distribution. Each of these is difficult to achieve. In all of these issues a great deal of attention is given to the design of mechanisms that can be implemented efficiently. Distributed systems built to date either do not address one or several of the essential properties of distributed systems or they are too inefficient to use.

The Amoeba Hardware Architecture

The Amoeba hardware architecture consists of four components: workstations, pool processors, specialized servers, and gateways. The *workstations* are intended to execute only processes that interact intensively with the user. The window manager, the command interpreter, editors, CAC/CAM graphical fornt-ends are examples of programs that might run on workstations. The majority of applications do not usually interact much with the user and are run elsewhere.

Amoeba has a *processor pool* for providing most of the computing power. It typically consists of a large number of single-board computers, each with several Megabytes of private memory and a network interface. A pile of diskless, terminalless workstations can also be used as a processor pool.

When a user has an application to run, e.g., a *make* of a program consisting of dozens of source files, a number of processors can be allocated to run many compilations in parallel. When the user is finished, the processors are returned to the pool so they can be used for other work. Although the poolprocessors are multiprogrammed, the best performance is obtained by giving each process its own processor, until the suply runs out.

It is the processor pool that allows us to build a system in which the number of processors exceeds the number of users by an ordere of magnitude or more. something quite impossible in the personal workstation model of the 1980s. The software has been designed to treat the number of processors dynamically, so new ones can be added as the user population grows. Furthermore, when a few processors crash, some jobs may have to be restarted, and the computing capacity is temporarily lowered, but essentially the system continues normally, providing a degree of fault tolerance.

The third system component consists of the *specialized servers*. These are machines that run dedicated processes that have unusal resources demands. For example, it is best to run file servers on machines that have disks, in order to optimize performance.

Finally, there are *gateways* to other Amoeba systems than can only be accessed over wide area networks. In the context of a project sponsored by the European Community, we built a distributed Amoeba system that spanned several countries. The role of the gateway is to protect the local machines from the idiosyncracies of the protocols that must be used over the wide area links.

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Why did we choose this architecture as opposed to the traditional workstation model? Primarily because we believe that the workstation model will become inappropriate in the 1990s, as it becomes possible to give each user 10 or 100 processors. By centralizing the computing power, we allow incremental growth, fault tolerance, and the ability for a single large job to temporarily obtain a large amount of computing power.

The Amoeba Software Architecture

Amoeba is an *object-based* system using clients and servers. Client processes use remote procedure calls to send requests to server processes for carrying out operations on objects. Each object is both identified and protected by a *capability*. Capabilities have the set of operations that the holder may carry out on the object coded into them and they contain enough redundancy and cryptographic protection to make it infeasible to guess an object's capability. Thus, keeping capabilities secret by embedding them in a huge address space is the key to protection in Amoeba. Due to the cryptographic protection, capabilities are managed outside the kernel, by user processes themselves.

Objects are implemented by server processes that manage them. Capabilities have the identity of the object's server encoded into them (the Server Port) so that, given a capability, the system can easily find a server process that manages the corresponding object. The RPC system guarantees that requests and replies are delivered at most once and only to authorized processes.

Although, at the system level, objects are identified by their capabilities, at the level where most people program and do their work, objects are named using a human-sensible hierarchical naming scheme. The mapping is carried out by the *Directory Service* which maintains a mapping of ASCII path names onto capabilities. The Directory Server has mecahnisms for performing atomic operations on arbitrary collections of name-to-capability mappings.

Amoeba has already gone through several generations of file systems. Currently, one file server is used practically to exclusion of all others. The Bullet Server, which got its name from being faster than a speeding Bullet, is a simple file server that stores immutable files as contiguous byte strings both on disk and in its cache.

The Amoeba kernel manages memory segments, supports processes containing multiple threads and handles interprocess communication. The process-management facilities allow remote process creation, debugging, checkpointing, and migration, all using a few simple mechanisms.

All other services (such as the directory service) are provided by the user-level processes, in contrast to, say, Unix, which has a large monolithic kernel that provides these srvices. By putting as mauch as possible in user space, we have achieved a flexible system, and have done this without sacrificing performance.

In the Amoeba design, concessions to existing operating systems and software were carefully avoided. Since it is useful to be able to run existing software on Amoeba, a Unix emulation service, called *Ajax* has been developed.

The VIEWS Project

Summary of the presentation given by Steven Pemberton†

Introduction

The Views project has developed from an earlier project at CWI, the ABC Programming Language. The purpose of ABC is to offer a powerful programming language and environment that is extremely easy to learn and use. As part of this aim, we wanted to reduce the number of different 'faces' that the user would see from the system: traditionally when programming, you must learn the command language of the computing system you are using, the command set of an editor, the programming language itself, how to use the compiler, and so on. With ABC, we managed to reduce this set to two: the language and the editor. ABC is not only the programming language, but also the command language, and for tasks that are not carried out with the ABC language, you use the editor: this includes actions like renaming and deleting objects, which is done by 'direct manipulation' of the objects involved.

It was quickly obvious that we could extend this method of working to more general tasks than programming alone, e.g., document management and process management, and thus was born the Views project.

Views, then, provides the user with a computing environment, which is characterized by

— what-you-see-is-what-you-get (WYSIWYG),

— direct manipulation,

— an open architecture,

— a consistent interface across applications.

WYSIWYG refers to systems where the system tries to keep the screen up-to-date with the true state of things. In fact Views is slightly stronger, leading us to coin a new term: TAXATA — things are exactly as they appear. While usually in WYSIWYG systems you are working with a copy of the object in some buffer (in other words, what you see is what you will get), in Views you are always effectively working with the object itself.

Open architecture means that it has to be easy to add new applications to the system, and that the user has much control and choice over individual aspects of the system and its interface (such as changing menus self, choosing which keys are available for 'short cuts', etc.).

User model

By the user model of a system we mean the model or set of rules which the user forms when working with the system. For example, if you work on a file, do you work on the file itself or on a copy? The basic user-model in Views is: every object in the system is editable, and every action is carried out by editing. Objects can be for example files, the clock, or menus. Traditionally, editing a file is performed on a copy, which then has to be explicitly written back to the disk. The Views model is closer to what one does in everyday life when changing a document.

As an example of the basic model, consider document management. Instead of individual commands to list the documents that you have in a directory or folder, to rename them, to delete them, copy them, and so on, you just 'visit' the folder — which is a document in itself in Views. This causes its contents to be displayed on the screen. To rename a document in the folder you just edit its name; to delete it, you just delete its entry; to copy a document, you just use the normal copy facility of the editor.

Similarly, to read electronic mail you just 'visit' your mail box. This causes its contents to be displayed as a list of message headers. Again, you can visit these individual messages (themselves documents), rename

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

them, delete them, copy them, all in exactly the same way.

An so on for other tasks: adding or deleting things to the printer queue, listing and deleting running processes, editing textual documents, or amending a spread-sheet.

Implementation model: invariants

The main characteristic of the implementation is that in general there are invariants between objects in the system. These invariants state that the contents of an object are a function of one or more other objects. For example, the profit for this year is the difference between the income and outgoings. If an object gets changed (usually by the user editing it), the invariant goes 'out-of-date', and has to be re-instated, which is done by calling a related function.

In fact, this mechanism is used very generally throughout the system, so that, for instance, displaying objects on the screen is done by application of the invariant 'the representation on the screen must match the object': if the object gets edited, then the screen gets updated. This means that the rest of the system can be completely oblivious of anything to do with output to the screen, or even that it occurs at all. In fact, all that an individual application sees is that its objects somehow change, and that it has to re-instate the invariants.

Present achievements

In 1989, the basic Views kernel was constructed. The idea is to construct the system incrementally, so that at all times we have a running system, and to add applications one by one.

The window interface was constructed on the basis of an earlier product of the ABC project, STDWIN, a window management package that allows programs that use windows to be portable between different windowing systems. For example, STDWIN already runs on top of X-windows on Unix, on the Apple Macintosh, and on the Atari ST.

A basic object can be represented in various ways. The basic display mechanism, which displays objects on the screen, was constructed for graphical objects as well as for textual ones, and the fundamental data-types as well as some primitive editing actions were implemented.

After these necessary preliminary steps, the invariant mechanism — probably the central part of the system — was built, and a start made on some applications, such as basic text-editing, file browsing, and message reading.

Plenary Discussion

chaired by Paul ten Hagen†

During this meeting the following problems and discussion issues were recognized:

1. Indexing
2. Editing & Authoring
3. Interaction:
   - Single Abstraction ↔ Fragmentation
   - User Model
   - Toolkit Approach
4. Multiple views (transformations of logical structure)
5. Languages: declarative, procedural, object-oriented
6. Reconciling problem oriented world model with data structure
7. ''Pixels, frames, bytes & bits, do not exist''
8. Expression in multiple media (film making, musical composition,..)

We will concentrate on indexing/editing/authorizing of video/sound/pictures.

INDEXING

ten Hagen: I propose we start with the indexing issue. How important is it that the system 'understands' the data in a medium?

Meertens: Not only do bytes not exist, but pictures do not exist. An image is just a map or presentation of the 'real' thing we are interested in. Image data are meaningless without a virtual interpretation. Images are as vague as the word 'data' itself. Indexing almost sounds like an issue of organization: how you organize things for efficient retrieval. But before we solve this we need to know what *questions* we are going to ask about the data.

Groen: I agree, but these questions depend on the application. For example in a medical database the tomographic images have meaning for the doctor who uses the system. Within such a specific framework we must and can find solutions to the indexing problem. Thus one can visualize a multimedia system as either a general-purpose but incomplete substratum for multi-device data, or as an application with knowledge-based interpretation and indexing.

Bulterman: It follows that text, sound etc do not exist either; they are only imperfect representations of the ideas they encode.

Meertens: I agree; this is why I disagreed with M. Kersten's claim that text indexing had been solved.

Bulterman: So it boils down to a question of what are the different representations for a data structure and what are the relations between those representations.

ten Hagen: An alternative possibility is indexing by example. For example we could select an image that represents something like the idea we want to find, and ask the system to look for images like our example.

van Geffen: Is indexing an inherently multimedia problem? Don't the same issues arise in single medium applications?

Blake: We can't really compare image indexing with text indexing. Bytes, pixels, frames etc are at a very low level i.e. the level of the sampling mechanism. To say that they don't exist just means that they are only samples of reality and have no physical meaning. Text characters are ''nonexistent'' only at the much higher level of meaning. In music we have a connection between

† CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

sounds and notes; there is no such simple connection between raster images and discrete picture objects.

Groen: The recognition of music from sound signals is a very hard problem. Fingerprint recognition and classification is solved and commercially available. One feels that there are many in-between problems where an initial reduction of the raster image to a vector line-drawing representation would make interpretation and indexing possible. However, image indexing in general implies image interpretation, which is very hard.

Lamming: Video and sound are temporal media: this makes them harder to scan. If there was no way for computers to search text, we could still resort to displaying it on a screen and having humans look through it. But there is no corresponding way for humans to rapidly look through hours of video tape. There are a few acceleration techniques for text, such as the use of keywords, sentence analysis etc but there is nothing for video. I guess there are lots of signal processing or image processing techniques that could be applied. An important observation is that for searching tasks we do not need to find a complete and exact solution. Any preprocessing that identifies likely segments of interest will save time. For example one can imagine hooking up a video tape server to an image processing system that is just instructed to measure frame-to-frame coherence; this gives us a guide to the location of scene changes, fast pans and so on.

Blake: An excellent example is optic flow techniques. If you can analyse a video stream into segments of roughly constant optical flow, this can provide many useful keys for searching and analysis.

Lamming: So you can decompose images into flow patterns? Can you give references?

Blake: The technique is to segment the image into pieces with roughly constant optic flow. Hilary Buxton at Queen Mary and Westfield College, London has done a lot of this work.

Groen: Another technique is to learn relationships by pattern recognition techniques without explicitly describing them. This can assist indexing.

ten Hagen: It was suggested by one speaker that we should confine ourselves to a single application. Is that sensible for the indexing problem? Is indexing only a problem for image processing and database people, or is it ubiquitous with user interfaces and so on?

van Geffen: Indexing doesn't interest everyone who is working in multimedia. For example, if I build up a computer aided instruction course, the script can use multiple media, but it is essentially canned and I know everything that is in it, so I don't need to index it.

ten Hagen: But if you store each pupil's responses on-the-fly, you will need to index them.

Bulterman: I see the major issues as being

— is multimedia inherently application-specific?

— is indexing representation-specific?

van Berkel: In art, presentation is extremely important as well as content. It is hard to define and distinguish these two.

Bulterman: One has to take a broader definition of content so that it incorporates not only the letter A but also the feeling or emotional message that you want to communicate.

van Berkel: I would not be so naive as to ignore such kinds of content; indeed I feel it is difficult to define and describe content.

Meertens: The issue is what questions we will put to our system. These questions will be distant from the precise visual presentation or data representation. For example, even an excellent painter probably cannot reproduce the 'Nightwatch' from memory without reference to the original.
Suppose you had a database of images of all 17th century Dutch masters. What questions do you ask such a system?

van Berkel: If I'm an artist, I ask it for Rembrandt's secrets. If I'm a historian, I ask it how they had parties in the 17th century.

Lamming: Database people would ask questions like 'how many paintings are there?', 'how many styles of painting?' and so on, questions related to the structure of the archive. The indexing and querying is a combination of content and representation.

Bulterman: An important issue is how we should focus our concerns. We can easily say we want a multimedia system to handle everything and know everything, but this is impossible to fulfil. It is a practical fact that (say) images need to be retrieved and displayed. Thus the pixels and bits must exist along with the other structures we want. This is not just a technological issue, but a fundamental one. The essential issue with multimedia is the time pressure. We could implement a multimedia system in Ingres if we had the patience to wait for huge images to be retrieved. Our problem is just to deal with the presentation efficiently alongside the other structures.

Lamming: It is not that pixels don't exist: it's just that database queries are not phrased in terms of pixels.

Bulterman: This is a prime example of how the abstract issues and the hard implementation need to be considered together.

van Berkel: There are plenty of examples of this in the real world.

Groen: You are tackling one of the hardest problems in information retrieval. For instance I have been presented with a problem involving the classification of images of Russian icons according to their composition, gestures and so on. I think this kind of problem is basically unsolvable.

Lamming: It seems as though there are demands coming down from the top and capabilities coming up from the bottom: some top-down problems like the Russian icons are clearly impossible, some bottom-up things like deblurring are clearly easy. Are there any meeting points between real problems and real solutions?

Groen: Yes. For example if you have a priori knowledge about a problem you may be able to select low-level criteria that are significantly associated with the higher-level questions you really want to ask. Then the low-level criteria can be combined to produce tailored algorithms that solve the real problem. If Russian icons could be classified by just looking at the height of a hand relative to a face, we could do the job.

Lamming: Yes, it is always easy to get carried away with finding exact solutions, when it would often be more useful to use approximate solutions to narrow down the range of material.

Blake: In artificial intelligence people have often tried to solve impossible problems, and it usually turns out that the best approach is to develop a partial solution and present the partial result to a human who makes the final decision.

Lamming: For example studies of human memory show that people often remember things (such as scenes in a movie) by their associations with completely different streams of information (such as the food they were eating at the time).

van Geffen: If you go ahead with your proposal for universal video taping, audio recording, text transcripts and so on, you will need to use a lot of such keys.

MULTIPLE VIEWS & USER INTERACTION

ten Hagen: Let's turn to the issues of multiple views and user interaction. There seems to be a difficulty in reconciling logical and representational views of the same information. One question is whether multimedia will make it harder or easier to work at the right level of abstraction.

In the early days of user interfaces, there were many slogans about transparency, but people moved away from this. Similarly most users of workstations usually communicate using one medium at a time, even though the system can respond in several media.

van Geffen: People do use several media at a time. There are user interface experiments showing that interactions through multiple media can be much more efficient than through a single (e.g. visual) channel. I agree there are problems in implementing and using this correctly.

ten Hagen: I know of counterexamples, such as speech input, which failed because people went back to using the keyboard.

van Berkel: The problem of balancing different media is also experienced by musicians who want to balance music and poem, or by film makers who want to balance the film with the music.

van Geffen: I agree, but what you are saying now is that you want to communicate a feeling or emotion with mixed media.

van Berkel: Well, sometimes I close my eyes in order to appreciate a poem better.

van Geffen: Yes, mixed media are not always optimal.

Lamming: Is it true that there are no bimodal user interfaces?

ten Hagen: Yes.

Lamming: For example is a combination of speech and typing inefficient?

ten Hagen: Some people just stop using their mouse and go back to the keyboard.

Lamming: I think we have actually taken a step backwards. In the old days when you started up a compilation on a PDP 8 it made lots of whirring and chugging sounds, and you knew exactly what the progress of your compilation was. Now your computer is silent, and we have no more progress cues. Maybe it's just that we do not yet have the right devices: perhaps we need sweat detectors to respond to the user's stress level.

ten Hagen: We could easily incorporate window tools that report the progress of a compiler.

Lamming: Computers spend most of their time in visual presentation, but this is actually a very narrow channel to the person. Other channels are under-utilized.

Groen: In robotics, the use of sensory information combined from very different sensors (smell, infrared,..) has had a lot of attention under the banner of 'sensor fusion'. The integration of these signals is pretty well understood.
It should be noted that some media interact more easily with certain others. For example it is easy to integrate mouse events into a simple vector graphics display. In this context, video presents the extra difficulty that it requires interpretation.

ten Hagen: Typically user interfaces present (a subset of) the input ingredients as a sort of vocabulary available to the user. There is no such vocabulary for images.

Lambert: I would really like to rotate the Mona Lisa to see a side view.

van Berkel: But you are not allowed to make her smile.

Groen: There is a Japanese multimedia project on the processing of music. Inputs can be scanned images of real music documents, typed music, or recorded audio. Outputs can be printed music, synthesizer programs, or sound. Is this what you would call multimedia, or is multimedia more about interaction?

ten Hagen: I would not make that distinction. Anything is interactive that can be represented. The multimedia repertoire is as rich as the repertoire of the machine.

EDITING

ten Hagen: Will the Views approach to syntax-directed editing still work when we move to multimedia? Can you maintain a uniform interface at the editor level? For example, artists will not want to be bound by the editor conventions, because presentation is as important as content.

van Berkel: Yes! One thing we need is a system that can catch 'events' in video sequences.

Bulterman: It should not be forgotten that there are simple tools and models available that can already provide this kind of manipulation, albeit in a rudimentary way. Thus we typically store films in video cartridges with a handwritten descriptive label. This provides an event-searching feature, although the granularity is poor! But rather than re-inventing such facilities, it may be useful for research purposes to find new ways to employ existing technology.

van Berkel: In case these representations are too large, how do you manipulate them?

Bulterman: Half an hour ago we said that representations were unimportant. Now it seems they are still important. My point is that there is a cross-over area that must be considered.

van Berkel: Sometimes you are simply unable to manipulate the data at all.

Bulterman: Sure, not all such problems can be solved usefully with simple technology. But filtering out the useful from the useless methods will be helpful.

Blake: You were saying that time pressure is the most important aspect of multimedia. Is it still?

Bulterman: Yes, but it is important not to get so caught up in the general and abstract problem that we don't recognize instances of our problem.

Blake: How do you fit multimedia editing into a standard editor framework?

Bulterman: Of course, there are lots of bad solutions. You can combine existing interfaces to produce the effect of combined cues. But the most interesting aspect is, if we constrain our breadth of view, how to support the manipulations we want.

PROGRAMMING LANGUAGES AND INTEGRATION

ten Hagen: There are many remaining questions, but let us concentrate on the issues of programming languages and integration. One can speculate about software tools. Will these be object-oriented techniques (or will multimedia provide a background for improving object oriented techniques)?

Blake: Object-oriented languages are fashionable, but indeed one very good idea of them is data abstraction. However it is interesting that if one starts to add constraints to an object-oriented language, one ends up with a declarative language. So you really want both types of language.

Lamming: I'm interested in hearing 3 minutes on what the CWI plans to build right now.

Bulterman: Different people will give different answers, but to sharpen the focus we are working on a multimedia messaging system, superficially an extension of ordinary e-mail, with which one can transmit mixed-media documents, and share them amongst users. The transmitted information contains not only the 'raw' data such as raster images, but also the logical structure of the document. Therefore this differs from facsimile and other technologies which impose no structure. In our system it will be possible to extract and edit the received document's contents and logical structure. The implementation is constrained by ethernet speed and workstation performance ceilings.

Lamming: So you would like to be able to mail a structured document comprising say Microsoft Word text documents, MacDraw and MacPaint images and other objects.

Bulterman: Yes, and the MacDraw-MacPaint structure information has to be preserved. In particular we need a bit rate high enough to support transmission of the structure information.

Lamming: To me, that is like ViewPoint. You would like to extend the ViewPoint concept to other media.

Bulterman: That would really be a construction project. What we want to do is to study the requirements of such a system, the logical structures needed, and so on.

Lamming: I'd be interested to know whether your multimedia work will include still images or audio as a first step.

Bulterman: Still images.

Lamming: Can you compare or contrast this project with 'andrew', 'diamond' or 'viewpoint'? Why don't you use one of these systems?

Bulterman: There is a fundamental aspect of the project that concerns group interaction. If we imported the systems you describe, we would not achieve one of our aims of getting different expert groups in the CWI to communicate at a high level. Some people would become experts and we would continue having lots of very good but rather autonomous researchers. We want to foster a general awareness of multimedia ideas and see what comes out of that.

Lamming: It is indeed difficult to get researchers to communicate well. I think this is an exciting project. You are about to build what we at Xerox call a 'sandpit' for new ideas. I think this is exactly the right approach.

ten Hagen: In your experience, how difficult is it to reorient people to communicate with each other, and to avoid their own private jargon?

Lamming: There is always a tremendous culture shock when someone moves to a new research institute, for example. I think the only solution is meetings like this where we can exchange views and organize joint work.
I would also like to suggest that CWI get in touch with

Bob Root at Bellcore (video, multimedia workshop in Boston, Mass)
Hewlett-Packard Bristol UK
Olivetti Cambridge UK
COIS group in Boston Massachusetts.

List of participants

E. Abbink
Océ Nederland B.V.
P.O. Box 101
5900 MA Venlo
The Netherlands
tel. +31 77 594048
fax +31 77 594313
email hjab@oce.nl
telex -

P. Cor Baayen
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4174
fax +31 20 592 4199
email pcb@cwi.nl
telex 12571 mactr nl

Adrian J. Baddeley
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4050
fax +31 20 592 4199
email adrianb@cwi.nl
telex 12571 mactr nl

Pierre van Berkel
University of Amsterdam
Faculty of Mathematics and Computer Science
P.O. Box 41882
1009 DBAmsterdam
The Netherlands
tel. +31 20 592 5135
fax +31 20 592 5155
email pierre@fwi.uva.nl
telex 10262 hef nl

Edwin H. Blake
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4009
fax +31 20 592 4199
email edwin@cwi.nl
telex 12571 mactr nl

Onno J. Boxma
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4094
fax +31 20 592 4199
email onno@cwi.nl
telex 12571 mactr nl

Dick C.A. Bulterman
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4147
fax +31 20 592 4199
email dcab@cwi.nl
telex 12571 mactr nl

W. Bulthuis
Philips Aachen IS/ICA
Weiβhausstraβe, P.O. Box 1980
5100 Aachen
B.R.D.
tel. +49 241 6003 617/615
fax +49 241 6003 709
email willem@ica.philips.nl
telex 17 241 4061

H. Peter Dijkhuis
CWI
tel. +31 20 592 4113

P.O. Box 4079
1009 AB Amsterdam
The Netherlands
fax +31 20 592 4199
email peter@cwi.nl
telex 12571 mactr nl

G. Dillema
P.T.T.
P.O. Box 15000
9700 CD Groningen
The Netherlands
tel. +31 50 821126
fax +31 50 122415
email -
telex 77088

F.J. van Drunen
Elsevier Science Publishers B.V.
Mathematics & Computer Science
P.O. Box 103
1000 AC Amsterdam
The Netherlands
tel. +31 20 5862 612/625
fax +31 20 5862 616
email SURF213
telex -

T. van Geffen
Océ Nederland B.V.
P.O. Box 101
5900 MA Venlo
The Netherlands
tel. +31 77 594048
fax +31 77 594313
email tvg@oce.nl
telex -

J.H.A. Gelissen
Philips Research Laboratories
W.B. 422
P.O. Box 80.000
5600 JA Eindhoven
The Netherlands
tel. +31 40 742689
fax +31 40 743350
email gelissen@neumann.prl.philips.nl
telex 35000 phtc nl / rout. nlw fvbi

Frans C.A. Groen
University of Amsterdam
Faculty of Mathematics and Computer Science
P.O. Box 41882
1009 DBAmsterdam
The Netherlands
tel. +31 20 592 5134
fax +31 20 592 5155
email groen@fwi.uva.nl
telex 10262 hef nl

Paul J.W. ten Hagen
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4133
fax +31 20 592 4199
email paulh@cwi.nl
telex 12571 mactr nl

J.W. van Hardeveld
P.T.T. Research
Neher Laboratorium
P.O. Box 421
2260 AK Leidschendam
The Netherlands
tel. +31 70 332 5771
fax +31 70 332 6477
email -
telex 31236 prnl nl

Ivan Herman
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4164
fax +31 20 592 4199
email ivan@cwi.nl
telex 12571 mactr nl

Martin L. Kersten
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4066
fax +31 20 592 4199
email mk@cwi.nl
telex 12571 mactr nl

Mik Lamming
Rank Xerox EuroPARC
61 Regent Street
Cambridge CB2 1AB
U.K.
tel. +44 223 341538
fax +44 223 341510
email Lamming.EuroPARC@Xerox.COM
telex -

Lambert G.L.T. Meertens
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4141
fax +31 20 592 4199
email lambert@cwi.nl
telex 12571 mactr nl

Sape J. Mullender
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4139
fax +31 20 592 4199
email sape@cwi.nl
telex 12571 mactr nl

Steven Pemberton
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4138
fax +31 20 592 4199
email steven@cwi.nl
telex 12571 mactr nl

Vincent Quint
BULL-IMAG
Unité Mixte de Recherche
2, rue de Vignate
Z.I. de Mayencin
38610 Gières
France
tel. +33 76 517879
fax +33 76 547615
email quint@imag.fr
telex -

Jos B.T.M. Roerdink
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 8020
fax +31 20 592 4199
email roe@cwi.nl
telex 12571 mactr nl

Bert P. Rouwhorst
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
tel. +31 20 592 4198
fax +31 20 592 4199
email bertr@cwi.nl
telex 12571 mactr nl

Carsten Wieschiolek
Detecon
Technical Centre Berlin
Voltastraße 5
tel. +45 30 46701 141
fax +45 30 46701 444
email -

Haus 10, Aufg. 2
1000 Berlin 65
B.R.D.

telex -