DESIGNING EFFECTIVE AND USABLE MULTIMEDIA SYSTEMS
IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP’s aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP’s mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP’s events range from an international congress to local seminars, but the most important are:

- the IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.
## CONTENTS

Preface
Programme Chairs and Programme Committee

### Part One Methods, Models and Tools

1. M. Wilson  
   Multimedia Design: from tools for skilled designers to intelligent multimedia design systems  

2. F. Garzotto, M. Matera, and P. Paolini  
   A Framework for Hypermedia Design and Usability Evaluation  

3. P. Pauen, J. Voss, and H-W. Six  
   Modeling of Hypermedia Applications with HyDev  

4. M. Apperley and R. Hunt  
   Design Support for Hypermedia Documents  

5. F. Nemetz, and P. Johnson  
   Developing Multimedia Principles from Design Features  

6. S. Morris  
   Media Transformations for the Representation and Communication of Multimedia Production Activities  

### Part Two Supporting Multimedia Design

7. K. Nakakoji, Y. Yamamoto, K. Sugiyama, and S. Takada,  
   Finding the 'Right' Image: Visualizing Relationships among Persons, Images and Impressions  

8. S. Phillips and J.T. McDonnell,  
   Structuring Multimedia data to facilitate decision making and reflection in product design  

9. J-W van Aaslst and C. van der Mast  
   Creating the multimedia project experience database  

10. M. Fjeld, K. Lauche, S. Dierssen, M. Bichsel and M. Rauterberg  
    BUILD-IT: a brick-based integral solution supporting multidisciplinary design tasks  

11. G. Herzog, E. Andre, S. Baldes, and T. Rist,  
    Combining Alternatives in the Multimedia Presentation of Decision Support Information for Real-Time Control  

12. B. Schonhage, P. P. Bakker and A. Eliens,  
    So Many Users - So Many Perspectives  

### Part Three Applications and Empirical Studies
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Authors</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Linda Lisle, Scott Isensee and Jianming Dong (Design review)</td>
<td>Developing a Multimedia Product for the World Wide Web</td>
<td>175</td>
</tr>
<tr>
<td>14</td>
<td>H. Eirund and G. Schreiber, A feasibility study for a mobile multimedia tourist guide</td>
<td></td>
<td>179</td>
</tr>
<tr>
<td>16</td>
<td>A. Roessler and V. Grantz, Performance Evaluation of Input Devices in Virtual Environments</td>
<td></td>
<td>205</td>
</tr>
</tbody>
</table>
Preface

Multimedia has become an important and established part of the computing industry as well as a topic of considerable academic research. As the multimedia marketplace becomes more crowded ease of use is becoming a key competitive advantage. Usability and effective communication are vital to ensure the success of multimedia designs and to avoid problem of information overloading. Multimedia is a topic which has defied succinct definition; however, a common theme in most multimedia research is investigating how advancing technology can be used to improve human computer communication beyond simple text and graphics based interfaces. Unfortunately, multimedia can be accused of being just hyped technology, so a prime motivation for organising this conference was to focus on how multimedia technology can be put to effective use. The conference objectives were to bring together researchers and practitioners from a variety of backgrounds to exchange current knowledge in the area, discuss design problems and solutions for improving product usability and shape future research agendas.

Multimedia systems are used in a wide variety of contexts although computer supported learning and entertainment have received most attention. These domains and the increasing diversity of other applications raise complex design issues. For example in educational applications sound design is necessary to promote learning by interaction and focusing the user's attention; while in decision support systems representing key information is important. Improving the design process and product quality implies the need for methods, models and support tools. As a precursor to methods we need to understand the design problem and develop sound theory-based principles and guidelines. Currently the literature on these topics is sparse.

These proceedings present the contributions to the IFIP 13.2 working group conference: "Designing Effective and Usable Multimedia Systems" held in Stuttgart, Germany on 9-10th September 1998. The papers from both researchers and practitioners describe design problems and solutions for improving product usability. In doing so they provide a variety of perspectives on design support, as well as advancing the understanding of usability issues and the design process for multimedia.

The keynote paper by Michael Wilson introduces the design problem of active tool support versus designers' knowledge and reviews three recent projects that provide design assistance as well as pointing towards progress in standardisation for
hypermedia design. The following five papers have a multimedia modelling theme. Garzotto et al describe their HDM modelling framework and then how usability guidelines can be organised by the framework for design assistance or usability evaluation. Pauen et al's Hydev multi-layered framework refines specifications of hypermedia from general domain models to presentation design at the instance level. This is followed by Apperley and Hunt whose HANDIE notation and design tool helps hypermedia specification with a focus on complex, composite documents and lists. Morris continues the design method theme by proposing a set of transformations for media selection and combination, based on a discourse model. Nemetz and Johnson argue for multimedia design principles grounded in sound theory or empirical evidence and propose an initial list based on a literature survey and conversational maxims.

The next three papers address diverse viewpoints on tool support for multimedia designers. Nakakoji et al describe a retrieval and traceability tool that matches aesthetic and affective descriptions of user needs (Kansei in Japanese) to appropriate multimedia materials; while Philips and McDonnell report application of multimedia design rationale (i.e. decision supported by diagrams, animations, etc.) applied to a case study in garment design. A survey of problems in the multimedia design process and creating a database to reuse this experience among designers and managers is proposed by van Aalst and van der Mast. These papers remind us that multimedia is a complex, multidisciplinary process that involves difficult issues of communication.

Automated plan-based design support is the theme of Herzog et al's paper that describes the presentation planner for decision support in traffic management domains. The planner is based on a discourse model of communication goals that enables automatic generation of multimedia interfaces for different user roles and tasks. The next two papers deal with collaboration from different viewpoints. First Fjeld et al report a virtual and augment reality tool that help designers in spatial configuration /planning tasks. Their system integrates manipulation of physical objects with virtual worlds. Schonhage et al's DIVA system uses visualisation and animation to support different user viewpoints for process model investigation and includes a high level presentation planner that utilises a media resource library.

The final group of papers describe applications and evaluations of multimedia products. Eirund and Schrieber's mobile multimedia tourist guide has novel script triggering so presentation is context-sensitive to the user's location. Isensee's et al web site illustrates many multimedia design problems as well as providing guideline advice. Duda's investigation of children's and adult's reaction to multimedia games demonstrates that the importance of fun and aesthetic appeal, while Roessler and Grantz's work illustrates that users' perceptions and performance with multimodal devices in virtual reality do not always agree.
Although these proceedings have collected a diverse set of stimulating papers on the multimedia design theme, it is worth reflecting on what we did not receive. Little attention has been paid to evaluation, although it is a concern for Garzotto et al. Design methods and models focused more on Hypermedia than multimedia, and multimodal dialogues were not covered. Media combination and presentation planning are dealt with but no recommendations were devoted to directing the user's attention in multimedia (see Faraday and Sutcliffe 1997, 1998). Empirical studies and theory based models are also absent, although Nemetz and Johnson's work is driving in that direction. Clearly there is much to be done, especially as multimedia user interface design standards are under development (ISO 1998). This conference has made a start in collating the sparse knowledge that does exist but clearly there is a pressing need for further research, and application of existing knowledge to advance current multimedia design practice (see Rogers and Schaife 1996). Finally we would like to thank not only the authors for the contributions but also the programme committee members for their effort in ensuring the high quality of these proceedings.

Alistair Sutcliffe
Juergen Ziegler
Peter Johnson


Programme Chairs

Alistair Sutcliffe, City University, UK
Juergen Ziegler, Fraunhofer Inst, Germany
Peter Johnson, University of London, UK

Local Organisers Franz Koller, Ruth Diessl, Joerg Nissler, Fraunhofer Inst, Germany

Programme Committee

Jim Alty, Univ Loughborough
Mark Apperley, Univ of Waikito, NZ
Elizabeth Andre, DFKI, Saarbrucken
David Benyon, Napier University
Ole Bernsen, Odense University
Khaled Choudhury, Bossard
Joelle Coutaz, IMAG, Grenoble
John Dowell, City University
Peter Faraday, City University
Ashok Gupta, Philips Research
Wendy Hall, Univ of Southampton
Lynda Hardman, CWI
Kristina Höök, SICS, Sweden
Alistair Kilgour, Herriot-Watt
Hilary Johnson, QMW
John Lee, Univ of Edinburgh
Andrew Monk, Univ of York
Martin Maguire, HUSAT
Mark Maybury, Mitre Corp, USA
Hari Narayanan, Auburn Univ, USA
Kumiyo Nakakoji, NAIST), Japan
Ian Mc Clelland, Philips Corporate Design
Fabio Paterno, CNR, Pisa
Klaus Haasis, MFG Medien
Peter Gorny, University of Oldenburg
Yvonne Rogers, Univ of Sussex
Matthias Rauterberg, ETH, Zuerich, Switzerland
Bob Spence, Imperial College,
Angela Sasse, University College
Neil Sandford, UK MM-SIG
Scott Isensee, IBM, Austin, USA
John Waterworth, Umea University
Jan-Willem van Aalst, Origin
Charles van der Mast, Delft University of Technology
Mike Wilson, RAL
Multimedia Design: from tools for skilled designers to intelligent multimedia design systems

M. Wilson
Advanced Hypermedia Systems Unit
Department for Computation and Information
CLRC Rutherford Appleton Laboratory
Chilton, DIDCOT, Oxon, OX11 0QX, UK
tel: +44 (0)1235 44 6619
fax: +44(0)1235 44 5831
email: M.D.Wilson@rl.ac.uk

Abstract

Multimedia design can be reduced to the process of choosing a presentation form which can be mapped to a set of domain concepts which you wish to communicate to users so that they can use the concepts to perform a task as effectively and efficiently as possible.

Since the design task is for multimedia, the set of possible presentation forms is as wide as possible, while there are constraints placed on the possible forms, and the mapping, due to cost, time, bandwidth of communication, presentation station abilities etc. derived from the overall task.

One of the major choices in multimedia design is to choose how much of the design process takes place off-line by a skilled human designer, and how much is performed automatically by the system. The consequences of this choice for the role of the designer and the concomitant interactions with the constraints on multimedia design are explored in this paper with reference to three systems developed in the last ten years: SMIL/GRiNS (Bulterman et al., 1998), MIPS (Jeffery et al., 1994; Macnee et al., 1995) and MMI2 (Binot et al., 1990; Wilson & Conway, 1991).

The Synchronised Multimedia Integration Language (SMIL pronounced smile) has recently been proposed by W3C for synchronising multimedia presentations over the world wide web, and GRiNS is the first editor to support authoring in it. SMIL supports four constructs: layout, timing, hyperlinking and tailorability of the presentation, while the human designer chooses the content of a presentation.
This is the most recent of the three exemplar systems, but also the one with least of the design process automated. The designer holds all knowledge of the task and domain, using it to describe the presentation using the four constructs provided by the language. The presentation is sensitive to available bandwidth, presentation station capabilities and user attributes which can be used at run-time to select between alternatives specified by the designer, but otherwise all decisions are made by the designer at authoring time. The designer has a view of which tasks the information may be used for, but it is really just information retrieval and presentation; the range of user domain tasks which a presentation may be used for is not limited by the designer or the system.

The control/navigation mechanism for the end user is also the most limited since hyperlinking is the only navigation available, and there is no stored dialogue state which can be used to relate to task structure, or tailor the presentation at the client.

The Multimedia Information Presentation System (MIPS) supported queries which were dispatched to heterogeneous information sources to retrieve multimedia information which was integrated into hypermedia presentations as answers to the query. In this case, a large part of the mapping that was design in GRiNS is ontology based query expansion & refinement and matching to database schema.

The media content of the presentation was retrieved from databases, but the layout, timing and hyperlinking and tailoring of content for design constraints were automatically constructed on the basis of the query. Compared to a SMIL/GRiNS presentation, the designer has a more remote role, since task descriptions, domain knowledge in the form of an ontology, and local dialogue state can all be stored in the presentation client and used to dynamically tailor the presentation at run time. The range of tasks which the system can be used for is limited by the domain knowledge to the tourism domain, and by the task knowledge to investigating and booking holidays. However, the task limitations can be overridden with a resultant degradation in performance of the query expansion process, and consequently in the information integration and design function. The control/navigation mechanism used in the answer is still limited to hyperlinks, although the query construction is based on a structured dialogue to elicit task, and user information which can later be used in the design process. As in a SMIL/GRiNS presentation, considerable attention is paid to the constraints of cost, time and security in using the communications layer to retrieve the content media items to be presented. The central storage of the ontology and metadata adopted in this system is impractical, but given the adoption by W3C of XML and RDF above that to describe metadata on the web, this approach may become practical in the near future.

The Multi-Modal Interface for Man Machine Interaction (MMI2) demonstrators support layout, timing, hyperlinking, tailoring of presentation, and both the design and construction of presentation forms from minimal basic elements automatically in order to achieve task goals. Here the designer has a minimal role compared to the
other two systems, since the entire presentation and dialogue is constructed at run time based on models of the domain, task, user and dialogue context which are used to guide the design knowledge built into the system. A consequence of the need for rich domain and task knowledge in the system, is that it is limited to the tasks for which these have been encoded. There is no graceful degradation when the limits of this knowledge is reached. Equally, the navigation/control of the presentation is most sophisticated here incorporating typed natural language (English, French and Spanish), direct manipulation of graphics, and the use of gestures as well as hyperlinks. But this is also domain and task limited due to lexica and planning systems. Although the application domain of the demonstrators was in network design and management, no consideration was given to networking constraints on the retrieval of information itself, although this is not a property of the approach. The earliest of the three systems, MMI2 results contributed to the Reference Model for Intelligent Multimedia Presentation Systems (IMMPS-RM) developed as an adjunct to the ISO Presentation Environment for Multimedia Objects (PREMO) standard activity (Bordegoni et al, 1997). This may result in the adoption of similar architectures for other intelligent systems in the future.

Each of the three example systems allows designers to produce interactive multimedia applications, to improve end-users’ task performance. Each tool operates over languages which represent the multimedia design, and each tool serves a role in an overall multimedia development method. The three systems clearly cover the spectrum from the central role of designers in SMIL/GRiNS through their partial involvement in MIPS to their peripheral role MMI2, as automation successively increases. In parallel with this, the representation of the content finally presented as media items becomes successively more abstract down this continuum from the raw assets and synchronisation information, through the raw assets and a logically represented query, to pure logical (and meta-logical, e.g. communication acts) representations. Equally, the control/navigation mechanisms for the end user become more varied and richer as one moves through the systems. It also appears that the task specificity of the systems increases as they depend more on abstract representations of content and control mechanisms. Each tool places different requirements on the skills of the designer: for GRiNS, they need graphic multimedia skills, and any analysis or representation they make of the task is up to them; for MIPS, the designer is not required to explicitly analyse and represent the task, although this improves system performance, but a representation of the domain ontology and metadata of the domain information sources is required - the multimedia graphic design skills are one stage removed here, being used to populate the information resources; in MMI2, analyses of the task, domain and user are mandated.

Clearly the multimedia design skills required of GRiNS are currently more available than those required for task and domain modelling. Equally, the interactive
multimedia applications developed in GRiNS can be applied to a wider set of tasks than those of the other systems. The enforcement of task and domain analyses in the development of the other systems leads to more richly interactive applications, but does it lead to more usable ones, or merely ones which are more easily evaluated, and therefore quality assure, over a known limited set of tasks?

ACKNOWLEDGEMENTS

The work reported in this paper was partially funded by the European Union Esprit programme under grants to the Chameleon consortium for SMIL/GRiNS, the MIPS consortium, and the MM12 consortium, as well as to the leveraging action to promote W3C in Europe W3C-LA, and under the Human Capital and Mobility programme to the TACIT Network.

REFERENCES


Part One

Methods, Models and Tools
A Framework for Hypermedia Design and Usability Evaluation

F. Garzotto, M. Matera, P. Paolini
HOC-Hypermedia Open Center, Department of Electronics and Information, Politecnico di Milano
Piazza Leonardo da Vinci, 32 - 20133 Milano, Italy
Phone: +39-2-23993520 - Fax: +39-2-23993411
{garzotto, matera, paolini}@elet.polimi.it

Abstract
This paper proposes a unified framework for the design and the usability evaluation of hypermedia applications. By providing a design model, a set of design guidelines, and a set of patterns of evaluation activities called abstract tasks, the framework helps a development team to perform both design and usability inspection in a systematic and cost effective way, and supports standardisation of activities and results across different designers and evaluators. The paper presents the framework and examples of its use, also reporting usability weaknesses detected in some commercially available hypermedia CD-ROMs.

Keywords
Hypermedia, Usability Evaluation, Hypermedia Design, HDM.

1 INTRODUCTION
It is generally acknowledged that the quality of a software product is strongly dependent from the quality of its design. In particular, design quality has effects on usability, a fundamental quality factor (Fenton, 1991) which concerns how easy is for users to learn a system, and how efficiently and pleasantly they can use it. We have explored the relationship ‘design-usability’ in a specific class of software products - hypermedia, and we have defined a unified framework that supports both the hypermedia design process and the usability evaluation activity.

The constituents of our framework are a hypermedia design model (HDM’98), a set of design guidelines, and a set of evaluation patterns for hypermedia usability called abstract tasks. The rational of our approach is the following. Design must be
supported by an expressive model (Garzotto et al., 1993), i.e., a language to describe the application constituents and to specify the design decisions, and by a set of guidelines which suggest how to achieve a good design. At the same time, the model identifies the 'subjects of interest' (Fenton, 1991) for evaluation, i.e., the application constituents which the evaluator should focus on; the guidelines suggest some usability properties of these constituents. The set of abstract tasks defines which operations must be actually executed on the application constituents to verify their usability.

In our approach, usability evaluation proceeds by inspection (Nielsen, 1993), i.e., it does not involve end users, but expert evaluators only. Although it is well known that the most reliable evaluation results can be achieved by combining inspection with user testing (Faraday et al., 1996), inspection techniques have the advantage that "... they save users (Nielsen, 1993)", do not require special equipment or lab facilities, and therefore are cheaper to use.

Finally, our framework distinguishes among different categories of design guidelines and evaluation tasks. Each category addresses design and usability of different dimensions along which a hypermedia application can be analysed: content, i.e., the actual information pieces stored in the application; structure, i.e., the organisation of the application content; navigation, i.e., the actual links and browsing mechanisms available to explore such structures; dynamics, i.e., the runtime behaviour of time-based media and links; user control, i.e., the operations available to the user to control the application dynamics; presentation, i.e., how all the above features are shown to readers (in other words, the visual properties of lay-out elements - buttons, windows, content fields, menus, etc.). So far, our framework addresses design and evaluation issues related to content, structure, navigation, dynamics, and user control; extensions to address presentation dimensions are subject to our on-going research.

The rest of the paper presents an overview of our framework, focusing on abstract tasks which are the most original aspect of our approach. Section 2 reports a short summary of the HDM'98 model. Design guidelines are briefly described in section 3. Abstract tasks are discussed in section 4, which also reports examples of usability problems detected with our evaluation framework in some commercial hypermedia CD ROMs. Conclusions and directions of our future work are described in section 5.

2 THE HDM'98 DESIGN MODEL
A primary component of our framework is HDM'98 (Garzotto et al., 1998b), the latest version of the Hypermedia Design Model HDM (Garzotto et al., 1993; Garzotto et al., 1994; Garzotto et al., 1995). For lack of space, in this paper we will only provide a short summary of the HDM'98 terminology, to help readers understand some terms frequently used in the following sections. For a discussion on the rationale of the various concepts, the reader is referred to previous publications.
In its current release, HDM'98 focuses on structural, navigational, dynamic, and user control ‘dimensions’ of hypermedia (as defined in the introduction) abstracting from presentation features.

Primitives for structural modelling distinguish between two sets of structures: hyperbase structures - which constitute the so called hyperbase layer (hyperbase for short) of the application, and access structures - which constitute the so called access layer. Hyperbase structures are used to represent domain information, while access structures provide entry points to the hyperbase. The hyperbase consists of entities and semantic connections among (parts of) them. Entities denote conceptual or physical objects of the application domain and are composite objects; their logical constituents are called components, and are organised according to some topological patterns (e.g., sequences, trees, lattices). Components in turn are made of nodes. Nodes are the actual containers of the multimedia data describing a component, and aggregate a number of content elements called slots. A node may correspond to a page, a page section, a full screen or partial screen window, depending on the adopted lay-out strategy. Their semantics is that different nodes of the same component describe different perspectives, i.e., different aspects concerning the component subject. A slot within a node can be static or dynamic, depending whether it stores time-independent media (such as formatted data, text strings, images and graphics) or time-based media (as video, sound, or animation).

The access layer consists of collections. A collection groups a number of members, in order to make them accessible. The members of a collection could be either hyperbase elements (entities, components, or nodes) or other collections (nested collections). A collection typically has (although it is not mandatory) a distinguished node called centre, which is informative about the collection content and is the starting point of the navigation within the collection. Members are collected according to some semantic criteria (e.g., in a museum application, ‘all paintings of a painting school X’, or according to an expected user’s goal (e.g., ‘the top ten paintings’ for a quick visit of the museum pieces). ‘Tours’ or tables of contents are modeled as collections in HDM’98.

Navigation primitives enable the description of browsing paths, i.e., links connecting nodes within the various structures. In HDM’98, links are of different categories: structural, applicative, or collection links. Structural links connect nodes within an entity according to its topology; applicative links connect nodes of different entities related by some semantic connections; collection links connect the constituents of a collection. If a collection has links connecting each member to another one in a given order, it is called guided tour. If a collection has links connecting centre with all members, and vice versa, it is called index. A guided tour index is a collection which includes both sets.

Dynamic primitives describe behaviour of dynamic slots and links. The behaviour of a dynamic slot concerns how its state evolves along the time by effect of user interaction, discussed below, or in dependency of the state of other slots.
HDM'98 provides a set of *temporal relationships* among slots occurring within the same node*. The behaviour of links refers to the effects of link traversing on the state of slots in the source and destination nodes. When a destination node is left and another is activated as effect of following a link, slots in the source (respectively, in the target) can be *paused* or *stopped* or *kept* playing, depending on the behavioural semantics of the link. The behaviour of links also concerns a mechanism sometime called *automatic navigation*. Automatic navigation means that the transition from a node to another one is performed automatically by the application, either by means of a time-out mechanism, or by synchronising the change of context with the execution of time-based media. For example, the transfer from node A to node B occurs when the audio comment on node A is over.

Finally, HDM'98 primitives for user control refers to the operations available to the user to control the behaviour of links (i.e., the effects of link traversing and automatic navigation - see above) and the behaviour of slots.

3 DESIGN GUIDELINES

The design guidelines proposed in our framework are empirical, in that they are founded on the personal experience of the authors and their group. We have designed, developed, and evaluated hypermedia applications for several years, for different companies and institutions, in a variety of domains; our guidelines try to capture the application properties that we consider useful to get well designed and usable applications.

Our design guidelines are organised in various categories - *structural guidelines*, *navigation guidelines*, *dynamic guidelines*, *user control guidelines*, and *content guidelines*, according to the multiple dimensions of a hypermedia that we have explored so far in our research.

The framework also includes two *meta-guidelines*: 'Be consistent' and 'Match the situation of use'. Consistency, one of the most general principles of good design, means that conceptually similar elements are treated in a similar fashion, while conceptually different elements are treated differently. The second meta-guideline corresponds to another general principle of good design, known as *task conformance* (Dix et al, 1993; Mayhew, 1992). Task conformance means that any design choice should take into account the physical and temporal context in which an application is used, the reason why users use the system, and their actual mental model. These rules are ‘meta’ with respect to all other guidelines since they can be

---

* The most important are *exclusiveness*, *disjointness*, *concurrency*, and *synchronization*. Two slots are mutually *exclusive*, if they cannot be active simultaneously. Two slots are *disjoint* if they can be active (and controlled) one independently from the other. Two disjoint dynamic slots are *concurrent* when they can be simultaneously active. Two slots are *synchronized* if they satisfy mutual temporal constraints (e.g., one becomes automatically active ‘after’ the other is de-activated).
applied to each dimension and property of an application, and are implicitly included within each guideline.

The guidelines we have defined so far are reported in the following tables. For lack of space, we will not discuss each guideline in detail, but will include only short comments or examples to clarify their meaning. The reader is referred to (Garzotto et. al., 1998b) for a more complete discussion.

Table 1 Guidelines for Structural Design

<table>
<thead>
<tr>
<th>S1</th>
<th>Define appropriate structures for the application content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The way of organising the hyperbase layer should be adequate to the size, the complexity, the semantics of the actual content. In the hyperbase, for example, if information about some domain objects are scarce, entities with a single component (in turn with a single node) are probably the best solution. On the contrary, a large amount of content should be better represented by entities structured in various components and nodes. Structure design and navigation design are strongly correlated, and this guideline should be considered in conjunction with N1 (see next table).</td>
</tr>
<tr>
<td>S2</td>
<td>Make access layer organisation 'complete' with respect to the hyperbase organisation</td>
</tr>
<tr>
<td></td>
<td>S2 addresses the access layer coverage issue, prescribing that each instance of each entity type should be a member of at least one collection. The rationale is that if an entity is mentioned nowhere in the access layer, users may never become aware of its existence until they traverse an applicative link (if any) taking to it.</td>
</tr>
</tbody>
</table>
Table 2 Guidelines for Content Design

<table>
<thead>
<tr>
<th>C1</th>
<th>Choose appropriate media, with appropriate ‘format’, to fit the content message of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>The designer of node content must choose the best format to convey the content message, considering the appropriateness of a medium or a combination of media, their physical features (e.g., as resolution, indicative size or duration), as well as rhetorical aspects, such as the literary style of text or the visual style of visual media.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C2</th>
<th>Make the content appropriate to the chosen delivery medium, its format, and the structure in which it occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>This guideline is the dual of the previous one. Once the node structure is defined, a node must be filled in with content which is coherent with such structure and with the physical and rhetorical format of the various slots.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C3</th>
<th>In collection centres, provide ‘correct’ information about collection members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection centres must support correct user’s understanding of what is in the collection, and how the collection is structured: i) the centre must store descriptors (text labels, icons, miniaturised pictures, or similar) to support the identification of all collection members, and only of them*; ii) the visual order of descriptors must corresponds to the navigational order among collection members. For example, if during forward navigation in a linear collection user finds a link ‘next’ from X to Y, the collection centre should show the titles of these two members, X and Y, one after the other, and not in a different order.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C4</th>
<th>When reusing a piece of information in a new context, adapt the portion of content which is strictly dependent on the original context</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a piece of content in a node depends on a given context, it should be removed when reusing the node in another context (and may be replaced with information needed by the new situation). For example, a textual reference in a node to the next node in a given collection must be removed when such a node is placed in a different context, where the following nodes may be different. This guideline is the companion, for content, of guideline N3 for navigation - see table 3.</td>
<td></td>
</tr>
</tbody>
</table>

* It might contain additional content, but member descriptors should be the primary content transmitted to the users.
**Table 3 Guidelines for Navigation Design**

<table>
<thead>
<tr>
<th>N1</th>
<th>Define navigational patterns appropriate for the topology of hyperbase and access structures, and for the complexity of the structures content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Links within and among hyperbase structures and access layer structures should be consistent with the topology of these structures. In linear entities, for example, we expect at least the links ‘next’ and ‘previous’ from a component to the following one and vice versa; still, additional links may also be useful for exploring a large component, e.g., ‘first’ and ‘last’ links to directly jump to the first or the last component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N2</th>
<th>Provide visible and efficient quit mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A general usability principle is to allow users to rapidly quit the application at any moment (Nielsen, 1994; Hardman et al., 1989); in hypermedia, this can be achieved by providing each node with an <em>easy understandable</em> quit command, or with a direct link to the place where such command is available. Most hypermedia applications provide the quit command only in one node (typically, the home page), but require many steps before reaching this context. In other cases, the quit function is not visible, and it requires to use platform specific shortcuts (e.g., ‘Alt+F4’ for Windows) not obvious for all users.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N3</th>
<th>When reusing the same structure in a new context, remove or modify links that are strictly dependent on a different context</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Consider, for example, the reuse of nodes across different linear collections. ‘Next’ and ‘previous’ links, from a node to the following and the preceding one, are strongly dependent on the actual collection and its linear order. Reusing the same node in another collection, with different members and a different order, requires to modify the destinations of such ‘next’ and ‘previous’ links. N3 is the companion, for navigation, of guideline C4 for navigation (see table 2.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N4</th>
<th>Support user perception of his/her current navigation context</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N4 is related to the ‘getting lost in the hyperspace’ problem - a typical usability issue for large hypermedia. To reduce the disorientation effect, N4 suggests that users should be always aware of the actual status of their navigation session, i.e., they should be able to understand their current position within the current entity or the current collection or the entire application. For this purpose, many hypermedia use active maps and overview diagrams, with indications of the user’s current location (and of previous steps), or some perceivable visual cues - for example, different page backgrounds of nodes to distinguish among different types of entities, or textual labels to indicate the title of the current entity.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N5</th>
<th>Keep backtracking facility distinct from hyperbase and access navigation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backtracking allows users to navigate, step by step, back to previous visited nodes. To avoid a potential source of disorientation, N5 prescribes <em>not</em> to provide backtracking commands in place of explicit navigation links, even in situations where their effects seem to be equivalent. For example, imagine that a user first navigates the entire structure sequentially, and then finds a way to jump directly to a node X from a node Y different from the one preceding X in the sequence. If ‘previous’ links are implemented by using backtracking, the use of this link from X returns the user to Y, and not to the node preceding X, as he or she would probably expect.</td>
</tr>
</tbody>
</table>
Table 4 Guidelines for Dynamics Design

<table>
<thead>
<tr>
<th>D1</th>
<th>Avoid behaviour interference among concurrent dynamic slots</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dynamic slots are concurrent when they are simultaneously active (see section 2). D1 prescribes that each disjoint slot should exhibit the same behaviour both when it is active individually, and when it is active concurrently with other disjoint slots, avoiding mutual interference and side-effects. The rationale for this guideline is that for building up a predictive model of how dynamic media behave (which is crucial for usability), users first try to understand how each medium behaves and can be controlled individually; then they experiment what happens when several media are simultaneously active. It is easier to recognise the sum of the individual behaviours of the various media, rather than to understand a new different behavioural combination.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D2</th>
<th>Define link behaviour appropriate for the link semantics and the content of source/target nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This guideline considers the effects of link traversing on the state of source and destination nodes, and on the behaviour of their dynamic slots. Dynamic slots in the source (or the target) might be reset to their initial state, or paused, or kept playing (see discussion in section 2.). The designer should consider a number of factors in order to decide which choice is more appropriate: the nature of dynamic slots, their duration, their content message, the combination of their states when links are traversed. For example, a sound slot in the source should be paused or stopped if link traversing automatically activates another sound slot in the destination, or if its semantic content is totally meaningless in the new context reached by navigation.</td>
</tr>
</tbody>
</table>

Table 5 Guidelines for User Control Design

<table>
<thead>
<tr>
<th>UC1</th>
<th>Provide user control on dynamic slots appropriate for the nature of their content, and for their format</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The commands designed for the user to manipulate the state of a dynamic slot depend upon various factors; among them, the nature of the slot (e.g., a picture can be zoomed in or out, but the same commands make no sense for a sound) and its physical properties such as resolution, size, duration-control commands such as 'start', 'stop', 'pause', 're-start', 'forward', 'backward' are meaningful, in principle, for all dynamic slots, but a video or a sound comment might require no interaction if they are very short. Ultimately, the degree of control must be appropriate to the actual need of users, based on their experience with digital multimedia and their goals in using the system.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UC2</th>
<th>Provide user control on automatic navigation appropriate for the content of structures and their size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This guideline refers to the user's ability of controlling the execution of automatic navigation (see section 2), e.g. suspending it, or switching from automatic to manual navigation, and vice versa. As for the control of dynamic media, the degree of control on automatic navigation depends upon various factors, such as size, content, and intended use of the navigation structure. Considerations similar to those mentioned for guideline UC1 can also be applied here.</td>
</tr>
</tbody>
</table>
4 ABSTRACT TASKS

Abstract tasks are patterns of operational activities that the evaluators should perform during the inspection in order to detect usability defects (Garzotto et al., 1998a). We use the term ‘abstract’, since: i) the activity specifications are formulated independently from a particular application, and ii) they refer to categories, or ‘types’, of application constituents more than to specific constituents.

Like the design guidelines, our abstract tasks are mainly empirical, in that they capture our experience on hypermedia product evaluation: they describe, using the HDM’98 vocabulary, what we do when we inspect a product for usability.

An abstract task is composed by five elements: the Title; the Focus of Action, i.e., a list of application constituents which are the focus of the evaluation activity; the Activity Description, i.e., what the evaluators have to do; the Intent, which is a short statement explaining what is the rationale of the abstract task, and which guideline(s) it refers to. It is important to note that, beside the evaluation activities explicitly described for the abstract tasks, there is an additional activity which is left implicit in the task formulation, although it is performed during (or after) the execution of each task. It concerns consistency checking: each abstract task has to be executed, in principle, on all the application objects of the category addressed by the task (mentioned in the ‘focus of interest’), in order to verify that conceptually similar elements have been designed and implemented in a consistent fashion across the application, and therefore show the same (good or bad) features. Sometimes consistency checking can not be accomplished exhaustively, especially for large applications. Therefore most times it is executed by induction: during an evaluation session, abstract tasks are applied only to a limited sample of objects, and the results are then generalised. The choice of the sample of objects might be difficult, and there is the risk of considering objects that do not show any problem, omitting other objects that might be more critical. From our experience, evaluators tend to start evaluation without choosing a priori such a sample; they just start executing abstract tasks on some random objects (the number of which depends on the evaluator’s personal style, the dimension of the application, and the intended duration of inspection). Then, they are induced to continue executing abstract tasks on additional objects if they find violations, with the intent of determining the severity of the detected problems on a larger set of situations.

Like design guidelines, abstract tasks are organised in various categories, according to the multiple dimensions along which a hypermedia can be analysed: structural tasks, content tasks, navigation tasks, dynamics tasks, user control tasks.

In this section, we will report a sample of abstract tasks, one for each task category. The reader is deferred to (Garzotto et al., 1998b) for a complete list, which currently amounts to thirty five abstract tasks. The tasks reported in this paper are the most representative of our approach, and those which helped us to discover the most frequent problems. For each abstract task, we will describe
examples of usability problems, detected on the seven commercial CD ROMs: *Art Gallery* (by Microsoft, 1993), a hypermedia guide to the National Gallery Museum in London; *Il Seicento* (by Opera Multimedia, 1995), an application about the European History of the XV century, whose content responsible is Umberto Eco; *La Pinacoteca Vaticana* (by E.M.M.E Interactive, 1996), an application about the painting collections of Vaticano, Rome; *Le Louvre* (by Montparnasse Multimedia, and Reunion des Musées Nationaux, 1994), a hypermedia guide to the paintings in the Louvre Museum in Paris that in 1995 won the 'best CD-ROM' award at MILIA’95 - one of the largest exhibition of multimedia titles world wide; *Musée d’Orsay* (by Montparnasse Multimedia, and Reunion des Musées Nationaux, 1996), a hypermedia guide to the paintings in the Musée d’Orsay, Paris; *The Italian Metamorphosis, 1943-1968* (by ENEL Italy, Progetti Museali, and Guggenheim Museum NY, 1994), which derives from an exhibition held at the Solomon R. Guggenheim Museum in New York.

4.1 Abstract task for structures

**Title:** ‘Coverage power of access structures’

**Focus of Action:** entity types + collections.

**Activity Description:** consider an entity type.

1. verify if there are collections which allow users to access its instances;
2. verify if there is at least one collection which allows users to access all its instances.

**Intent:** to verify the completeness of the application entry points, i.e., if the access structure efficiently supports the access to the hyperbase entities (see guideline S2).

**Detected Problems:** the application *Musée d’Orsay* has three hyperbase entities: ‘Painting Collections’, ‘Exhibition Rooms’, and ‘Painters’. What we noticed is that there are no entry points for the entity ‘Painters’. The top level index allows users to access only the entities ‘Painting Collections’ and ‘Exhibition Rooms’. Moreover, there are no collections including the instances of the entity ‘Painters’. The only way to access this entity is to navigate in the hyperbase, i.e., to follow applicative links from the instances of the entity ‘paintings’.

4.2 Abstract task for content

**Title:** ‘Accurateness of the information content in collection centres’

**Focus of Action:** collection centres.

**Activity Description:** verify if the information content of a collection centre accurately describes the content of the collection. For example:

1. verify its correctness, i.e., if the supplied descriptions correspond to the actual content of the collection;
2. verify its completeness, i.e., if it gives indication about all the members in the collection;
3. verify its ordering, i.e., if the order in which collection member descriptors are visually listed in the centre corresponds to the navigation order among collection members.

*Intent:* to verify how well the centre of a collection supports users' understanding of what is and what is not in the collection (see guideline C3).

*Detected Problems:* in *La Pinacoteca Vaticana*, there are several collections (corresponding to different painting taxonomies), that have a centre presenting some thumbnails, one for each painting belonging to the collection. A click on a thumbnail allows users to enter the collection, and to get to the painting node. Starting from there, it is possible to navigate both forward and backward (two buttons are provided). What is surprising is that such navigation follows an order which is exactly the opposite of the one suggested by the collection centre. Therefore, in each collection member, the 'next' (respectively 'previous') button leads to the previous (respectively next) painting displayed in the collection centre.

### 4.3 Abstract tasks for navigation

*Title:* 'Complexity of applicative navigation patterns'

*Focus of Action:* applicative links.

*Activity Description:* in an applicative link:

1. navigate from the source node to one of the target nodes;
2. randomly visit one of the target nodes;
3. systematically visit all the target nodes;
4. every time a target node is reached, try to navigate back to the source node, without using backtracking commands.

*Intent:* to verify if an applicative link has a navigation pattern which is appropriate for the semantic relationship it represents, and if it includes symmetric links from the target nodes to the source nodes (see guideline N1).

*Detected Problems:* in *Art Gallery*, by executing this task on several applicative links, we discovered instances of the same link type that are symmetric, and other instances that can be traversed only in one way. There is a link, for example, from *Tempera* to *The Baptism of Christ* (*Tempera* is the technique used for that painting), but there is no reverse link from *The Baptism of Christ* to its technique.

*Title:* 'Visibility of navigation status in collection navigation'

*Focus of Action:* collections.

*Activity Description:* in a collection, access an arbitrary member, and identify its position in the collection structure.

*Intent:* to verify if members of a collection contain clear indications about their location in the collection, so that to support users' orientation (see guideline N4).

*Detected Problems:* in the application *Il Seicento*, each covered 'topic' is presented as a 'book chapter', and organised in a sequence of 'pages', with two distinct buttons for going back and forth.
In the pages, there are no presentation elements that help to identify which point of the sequence has been reached, but a little icon, representing a stack of sheets which changes, adding or removing one sheet after the users moves two pages forward or backward. In our opinion this is a poor and not much visible mechanism for representing the navigation status, especially if compared with the mechanisms provided in other applications, where an explicit label indicates, for each navigational step, which member has been reached, how many members have been already visited, how many members are left.

4.4 Abstract task for dynamics (media and links behaviour)

*Title:* ‘Link behaviour & dynamic slots’

*Focus of Action:* dynamic slots + links

*Activity Description:* consider a dynamic slot:

1. activate it, and then follow one (or more) link(s) *while the slot is still active*; return to the ‘original’ node where the slot is placed, and verify the actual slot state;

2. activate the dynamic slot; *suspend it*; follow one (or more) link(s); return to the original node where the slot has been suspended and verify the actual slot state;

3. execute 1 and 2 traversing different types of links (both to leave the node and to return to it);

4. execute 1 and 2 by using only backtracking to return to the original node.

*Intent:* to verify the cross effects of navigation on the behaviour of dynamic slots, i.e., what happens when the activation of a slot is followed by the execution of navigational links and, eventually, backtracking (see guideline B2).

*Detected Problems:* in the *Louvre* application there are nodes of type ‘Painting Presentation’, which show a full screen painting image with an audio comment. By applying this abstract task on these nodes, we noticed that the audio is interrupted when the user navigates to another node. In other nodes, those of type ‘Loupe’, we discovered instead that, if a link is selected while animation and audio are still active, the audio comment continues till the end of the current audio ‘slice’, although the current node is immediately replaced by the link destination node. Thus users finds themselves on a content which has nothing to do with the comment they are listening to. What is even more surprising is the following: if the selected link takes to a different entity, any further click anywhere on the destination node interrupts the play of the current audio slice, but if the link is structural, i.e., takes to a different component of *the same* entity, any further click does not interrupt the audio slice.
4.5 Abstract task for user control

Title: ‘Complexity of control on automatic guided tour navigation’

Focus of Action: collections or entities with automatic navigation.

Activity Description: in an automatic guided tour:
1. verify the complexity of control on the automatic navigation, in terms of number and type of control commands. For example, suspend the automatic navigation and restart it, or suspend the automatic navigation and proceed manually in the collection navigation, etc..
2. verify if the set of the control commands is appropriate, in accordance with the collection structure and organisation.

Intent: to verify the appropriateness of the commands for controlling the automatic guided tour navigation (see guideline UC1). This task is the analogous, for navigation, of the control task defined above for dynamic slots.

Detected Problems: in the application *Italian Metamorphosis, 1943-1968,* entities are organised as linear sequences of pages (nodes), each one containing three synchronised media: a scrolling text, a slide show of images, and a audio comment. Navigation along these pages is automatic, and starts as the entity is entered. The transition from one page to another occurs automatically at the end of each sound comment. The only available command to control the automatic navigation is ‘STOP’, which interrupts the sound command, and abruptly takes the user to the last page. There is no way to restart the activation, unless the user is willing to play the ‘usual’ trick of navigating somewhere else, and then start the navigation again. This behaviour is consistent across the application. Unfortunately, the lack of control is disturbing, and the effect of the stop command is not self-evident: the users find themselves on a totally new page (the last one), and might get disoriented.

5 CONCLUSIONS

The intended users of our framework are mainly hypermedia design and usability specialists, but they can also be software developers or practitioners. The framework can be used in several stages of the hypermedia development process: model and guidelines are useful during the design phase, abstract tasks during design evaluation, prototype evaluation, and final product evaluation. The output of the design phase is a HDM’98 specification of the application schema, i.e., the types of hyperbase and access structures, their behaviour, and the user operations available on the various types of objects. The output of an evaluation is an organised list of potential usability problems, classified according to the various categories of objects and features of the application.

The use of a framework like the one proposed in this paper has several advantages. It can be used to approach the processes of hypermedia design and evaluation more systematically and efficiently; it can improve the communication among the members of the development team, by providing a common vocabulary
of concepts, terms, and principles; it can support standardisation across different designers and evaluators.

So far, our framework has not addressed design and evaluation of presentation issues, i.e., all features of an application which concern lay-out objects and their properties. Although most presentation rules can be defined in terms of conventions, standards, generic principles for user interface design, which can be found in the HCI literature (Dix et al., 1993; Mayhew, 1992; Preece, 1994), it is also true that we need presentation models, guidelines, and abstract tasks that address hypermedia specific features (e.g., anchor visualisation). Together with the investigation of additional guidelines and abstract tasks for content design and evaluation, presentation issues are the subjects of our current activity to complete the framework.

A further aspect, not addressed by our framework yet, has to do with rating the severity of the detected usability problems. This is necessary in order to prioritise the activities needed to fix the problems, and to avoid expending disproportionate effort on low-priority problems. Severity ratings are derived from an estimate of the expected user impact of each usability problem, as well as budget issues. Identifying criteria for severity ratings is one of the directions of our future work. A related direction of future research concerns defining a more precise mapping between design guidelines and situations of use. We need to relate hypermedia specific categories of user tasks, application domains, contexts of use, to the applicability of design guidelines.

Finally, we are planning to validate the overall framework, currently based on our long-term personal experience, by performing experiments involving users.

6 ACKNOWLEDGMENTS
This work has been partially founded by the European Commission Project 'SITMOON', and the CNR Project 'Museo Virtuale dell'Informatica'.

We acknowledge Prof. M. F. Costabile, from Department of Computer Science, University of Bari, Italy, for her valuable suggestions.

7 REFERENCES


7 BIOGRAPHY

*Franca Garzotto* is Associate Professor of Fundamentals of Computing at the Department of Electronics and Information, Politecnico di Milano. Her research interests include hypermedia modelling, authoring, evaluation, and multimedia for cultural heritage and education. She has been involved in various European research projects in the above fields.

*Maristella Matera* is a Ph.D. student at the Department of Electronics and Information, Politecnico di Milano. Her research interests span visual interfaces for databases, intelligent multimedia presentation systems, hypermedia modelling, usability engineering, usability testing.

*Paolo Paolini* is Full Professor of Multimedia and Computer Graphics at the Department of Electronics and Information, Politecnico di Milano, where he also serves as Scientific Director of HOC (Hypermedia Open Center). His research interests include hypermedia design, hypermedia development systems, WWW interfaces to data bases. He has been scientific responsible of various European research projects in the above fields. He is currently Associated Editor of ACM Transactions on Information Systems (TOIS).
Abstract
This paper introduces the \textit{HyDev} approach to a structured and systematic development of hypermedia applications. \textit{HyDev} focuses on the early phases of the development process, i.e. analysis and design. The requirements and key aspects of the software to be built are captured with tightly coupled description models. The main emphasis of this paper lies on the models for the requirements engineering phase which, simply spoken, capture structure, content and presentation of a hypermedia application at an appropriate level of abstraction.

Keywords
Hypermedia, multimedia, authoring, software engineering, requirements engineering, modeling, model-based software development, \textit{HyDev}

1 INTRODUCTION

In the last years importance and distribution of hypermedia applications - in the following abbreviated as HMA - have significantly increased. The range of HMAs, e.g. electronic books, multimedia learning/training software as well as product catalogs and presentations, is very inhomogeneous. Some of these primarily appear as documents, others can be more appropriately characterized as complex software systems.
There is also an increasing interest in development methods. However, traditional software development methods are mostly inappropriate - HMAs have several characteristics which distinguish them from conventional software. Probably most noticeable is the multimedial representation of the application's information and objects: In addition to text HMAs can contain graphics, pictures, and elements with a temporal dimension, like audios, videos, and animations. Secondly, HMAs are characterized by special elements and structures. For example, the flow of what is presented or happens can be determined by a kind of film-script. Typically, this can be observed in guided tours or interactive comics. In other words: HMAs can have a narrative structure. Further special elements are agents and 2D-/3D-objects. Another difference refers to user operations: Most important is navigation, i.e. is the selection of objects to be presented. Less important is typical information processing like the creation of new objects, computation, and object modification.

HMAs usually are of a considerable complexity. For example, common computer games often have a complex inner structure and audiovisual organization as well as a high degree of interaction. Therefore, specific development methods with suitable milestones and documents are principally advisable and conducive in the domain of hypermedia applications.

Established software engineering methods work well in areas like data processing, engineering or telecommunication software but can not be directly applied to multimedia applications. There are a few specialized approaches to a systematic development of HMA (see chapter 3). But these are still too immature and can not handle the complex structure of HMA at an appropriate level of abstraction. Commercial authoring tools, e.g. macromedia’s Director, concentrate solely on the implementation and do not offer support for the early phases of the development process, i.e. analysis and design.

For these reasons HMA development is usually quick&dirty, resulting in low correctness, robustness, and maintainability of the end products. Consequently, practitioners like for instance Kathy Kozel (Kozel, 1996) complain about the negative consequences for the practical development work.

On the grounds of these observations we have developed HyDev*, a domain-tailored approach to structured and systematic development of HMAs that explicitly takes the above mentioned characteristics into account.

The rest of this paper is organized as follows. The second chapter gives an overview over HyDev and the activities in the various phases of the development process. In chapter 3 we look at related work in this field. As our main contribution we subsequently introduce HyDev’s three requirements models in more detail. Chapter 4, 5, and 6 deals with the domain model, the instances model resp. the representation model. The paper concludes by summarizing our contributions and discussing future work to be done.

*The acronym HyDev is made up of the words hypermedia and development.
2 THE HYDEV APPROACH

2.1 Overview

HyDev works with several distinct models with fine-grained relationships between model elements. The various models build on one another and are the cornerstones of the development in that they capture certain aspects and decisions which are of great importance for the system to be built. Each model views the application under development from a different perspective, has a certain level of abstraction, and is intended for a specific development phase. In this respect HyDev is a model-based approach.

2.2 Requirements analysis

In contrast to conventional software development requirements analysis for hypermedia applications mainly deals with content and quality of presentation. Requirements resulting from the system’s environment and the users’ work context are much less important. Nevertheless, the structured development of an HMA should have an explicit requirements engineering phase in which the main features of the system to be built are identified and documented in a non-technical form. Mastery of the complexity succeeds primarily by abstraction, i.e. limitation to selected important aspects.

In this context it is of high importance that requirements engineering level documents do not anticipate the actual implementation. In particular, there should be no assumptions concerning the media objects. For example, it should not be necessary to know which particular videos or audios will be integrated into the final HMA. Quite important also is that an approach does not force the developer to use a specific authoring tool, nor should it restrict the developer concerning the choice between an authoring tool or a special hypermedia program library.

Roughly spoken, the aspects captured by HyDev requirements analysis models are structure, content and presentation of the hypermedia application. The models are called domain model, instance model and representation model, respectively. An important issue in choosing modeling concepts is to appropriately deal with the document-software dichotomy. HMAs are both: documents with individually presented objects as well as software systems which present and manipulate uniformly structured objects. In the following we briefly characterize the three models.

The need for an instances model

The navigational structure of an HMA is mainly determined by its content, i.e. the objects to be presented. We have observed that in this respect instances and their relationships are at least as important as the underlying classes and their relationships. As a typical example we consider the development of a CBT* course. There one must decide in detail which subchapter a certain chapter has, which references
are therein and which examples and assignments are included. So, in contrast to conventional software it is not sufficient to simply employ an ER-model or an OOA-model. Rather, the application's underlying objects have to be explicitly considered in a separate instances model. Chapter 5 contains more details about the instances model.

The need for a domain model

The consideration of objects is only conducive if corresponding classes and their relationships have been carefully modeled beforehand. Therefore like with conventional software a class model - also called domain model - is necessary. Such a class model has to be tailored to the special kinds of objects that we can find in HMAs (e.g. narrative units, agents, 2D-/3D-items). However, this kind of domain modeling, e.g. in the form of an adapted or extended variant of OOA, is entirely missing in existing approaches. The details about the domain model and its special kinds of classes can be found in chapter 4.

The need for an representation model

On the basis of a domain model and an instances model it is not appropriate to proceed directly with the implementation using concrete media objects. Obviously, domain and instances models leave a considerable degree of freedom for design decisions. We believe that it is essential to specify these aspects during the requirements engineering phase. But for two reasons it is advisable to do this at a higher level of abstraction. On the one hand the early specification of details such as position, layout, color and timing would anticipate the actual implementation and result in a unnecessary high effort of revision. On the other hand such abstraction helps mastering complexity.

Thus, simply spoken, it is specified in which way objects are presented to the user by the running application, i.e. as text, graphic, video, vrml-world or the like. Further specifications concern the navigational structure as well as user interactions. A specific model is needed for such aspects of the object representation. Another argument for this model is that logical structures between objects do not correspond directly to structures between object representations. It is very well possible that an object has several different representations. Conversely, several objects may have one common representation. Chapter 6 is devoted to the representation model.

2.3 Specification & design

During the specification & design phase the requirements and decisions captured with the three analysis models are concretized and refined. This way one finally obtains a complete specification of the system to be build. The decisions of this phase concern inter alia:

*CBT = Computer Based Training
• user interface objects (buttons, menus, ...)
• details of the temporal and spatial relationships between representations
• playback parameters
• media objects and playback effects
• playback channels
• details of the interaction techniques
• quality of service
• requirements on the underlying hardware of the target system

2.4 Implementation

The results of the specification&design phase serve as a starting point for the implementation phase. Among the activities of this phase are:
• creation of the media objects
• actual implementation and realization (e.g. with the help of an authoring tool)
• programming
• realization of effects
• adaption to the hardware of the target machine

3 RELATED WORK

In this chapter we take a look at selected other methods for structured hypermedia design and work out differences between these and HyDev. We will restrict ourselves to the Relationship Management Methodology (RMM) (Isakowitz et al., 1995) and the Object-oriented Hypermedia Design Model (OOHDM) (Schwabe and Rossi, 1995). Other modeling approach in the hypermedia field, like Dexter (Halasz and Schwartz, 1994) and AHM (Hardman and Bulterman, 1994), offer technology independent modeling concepts for hypermedia documents. They do not deal with development methods and rather closely stick to the document paradigm.

RMM is based on the Entity-Relationship model. The first step in the development process is a conventional ER diagram which captures the information domain of the application. This is followed by the definition of so-called slices, i.e. meaningful groups of an entity's attributes. The result of this step is an enriched ER diagram, the ER+ diagram, containing entities, relationships, and slices. Next the navigational design is developed. All navigational paths are derived from relationships between entities. They are specified in terms of entity properties and relationships. This step results in the so-called RMDM diagram, the cornerstone of RMM. The following steps comprise the conversion protocol design (each element of the RMDM diagram is transformed into an object in the target machine, for example a listbox) user-interface design (i.e. the design of screen layouts for every object of the RMDM diagram), and runtime behavior design (dealing with aspects such as link traversal, backtracking, and navigational mechanisms).
OOHDM is a model-based approach that comprises four main activities: conceptual design, navigational design, abstract interface design, and implementation. During conceptual design an object-oriented domain model with classes, relationships, attributes, and subsystems is defined. OOHDM views an HMA as a navigational view over the conceptual model. The navigational structure is defined by a schema specifying navigational classes such as nodes, links, and access structures. While nodes represent views on conceptual classes, links are derived from conceptual relationships. In the abstract interface design phase an abstract interface model is built. It captures which interface objects the user will perceive, the way in which navigational objects will appear, how navigation is activated, and synchronization aspects.

Both RMM and OOHDM use plain class models which do not take into account that HMAs can have special elements like narrative structures, spatial objects or agents. Therefore, they have to model such elements in a more complicated and less comprehensive way. Apart from that, they work only with an ER model resp. a class model. The consideration of individual objects is left out. Consequently, representations of concrete objects do not occur. We consider this a significant shortcoming. For example, we believe that many navigation connections only make sense with object representations. RMM and OOHDM appear to be most suitable for applications with uniformly structured data like, for instance, product catalogs. More complex types of HMAs, like ingenious games, will require more sophisticated specification techniques.

4 HYDEV'S DOMAIN MODEL

The idea behind HyDev is that an HMA is based on a collection of various objects and (potentially complex) relationships between them. In the scope of the domain analysis the corresponding classes and relationships are specified in a domain model. Since HMAs can have special elements that are usually not found in conventional software, HyDev's domain model works with the following specialized classes in addition to conventional classes:
- classes of narrative structuring units: \textit{N-classes}
- classes of objects with a spatial dimension: \textit{S-classes}
- classes of agents: \textit{A-classes}

These specialized classes can have attributes and operations and can be connected by associations, inheritance-, and part-of-relationships just like conventional classes. Additionally, there are numerous specialized relationships (described below in the context of according classes). By these means, HyDev allows a much more adequate domain modeling than a conventional OOA or ER model.

The notation of the domain model follows UML (Booch et al., 1997). Therefore, associations are notated by simple annotated lines between classes, and part-of- and inheritance-relationships by lines with rhombuses resp. arrows. The specialized re-
lationships are marked with special symbols (see below). In order to make the differ­
ent kinds of classes distinguishable, the symbols for the specialized classes have a
small signet in the upper left corner: a ~ for N-classes (symbolizing flow), a 0
for S-classes (symbolizing a spatial cube), and a ʌ for A-classes (symbolizing a
man).

4.1 Classes for narrative units (N-classes)

In many HMAs, especially sophisticated games, one can find a narrative structure.
For example, the interactive animated online comic Madleine's mind (Madmind,
1997) is organized in acts, episodes, scenes, and steps. A narrative structure is based
on special objects that have the character of film-scripts. These objects are called
narrative units. They are used for modeling the flow of what is presented or what
happens. Narrative units structure an HMA concerning its thematic contents by
rouping elements on the grounds of narration. It is possible that multiple narrative
units constitute complex narrative units. They are modeled by N-classes.

Among the specialized relationships between N-classes are:
• the sequence-relationship by which the narrative or logical sequence of narrative
units, i.e. the narrative structure is modeled. For example, in an animated multi­
media comic the various scenes follow each other.
• the simultaneity-relationship for modeling narrative simultaneity of narrative
units. For instance, such a relationship can be used to express that two parts of a
story happen at the same time.
• the prerequisite-for-relationship that allows to specify that one narrative unit is a
prerequisite for one or several other narrative units. For example, a flashback can
be necessary for the comprehension of subsequent episodes.

Beyond that there is a participate-in-relationship that can exist between N-classes
and the other kinds of classes. It aims at the objects that participate in a narrative
unit. For example, a scene can take place in a certain room in which some agents
appear; the room and the agents are then participants of that scene.

As an example we look at an animated multimedia comic. The comic might con­
sist of several storylines which can happen simultaneously. Each part consists of ep­
isodes and maybe a flashback. Episodes follow each other or a flashback. A
flashback can be a prerequisite for an episode. Both consist of scenes. The next pic­
ture shows the corresponding N-classes along with their attributes and the relation­
ships. Since flashbacks and episodes are very similar the model has a common
superclass for them (Block).

4.2 Classes for objects with a spatial dimension (S-classes)

HMAs often have objects that are characterized by their spatiality, for example
rooms or 2D- or 3D-objects within rooms. These objects with spatial dimension are
Figure 1  N-classes of a comic application.

often found in games, animated interactive comics or virtual reality software. For modeling of these objects HyDev’s domain model has S-classes. The adjacent-relationship or the contained-in-relationship for instance are among the many special relationships between S-classes.

Objects with a spatial dimension can have a specific dynamic behavior. A good example are pyrotechnic articles that can be viewed in a digital product catalog for fireworks. Pyrotechnic articles are objects that have a specific behavior: They have a certain flight behavior and produce certain light and sound effects. Such aspects are modeled with the help of N-Classes or - analogous to attributes and operations - with simple behavior descriptions as part of the corresponding class definition.

In the next picture an extract of the corresponding domain model can be seen. As there are two kinds of pyrotechnic articles, PyrotechArticle has two subclasses: Rocket and FireCracker, each of which defines additional special attributes. According to the remarks above, the class definition for PyrotechArticle has three subsections for the specification of attributes, operations, and behavior. For modeling the complex orchestration of fireworks we include an N-class Orchestration. It is connected with the class Fireworks via another special relationship not mentioned yet, the arrangement-relationship. Participants in such an orchestration are rockets and fire crackers. Therefore, Orchestration has a participate-in-relationship to PyrotechArticle.
Finally, HMA s - especially games and virtual reality applications - can have elements which are characterized by some kind of independence and autonomy. These elements are called agents. They are modeled with A-classes. Typical agents are characters (e.g. in adventure games) or guides through virtual worlds. Agents never stand alone for themselves but always participate in narrative units. They also can have a certain behavior which is then modeled analogous to the behavior of objects with a spatial dimension. The behavior is the expression of their autonomy. Agents can have certain tasks, pursue an aim (within limits), react to modifications of their environment, and interact with each other.

Let's consider a typical tactical game with a hero and opponents such as helicopters and tactical groups consisting of soldiers. Obviously, these are agents that are to be modeled by A-classes. The classes Helicopter and TacticalGroup have a common superclass Opponent. Both the hero and his opponents participate in a battle which is modeled by an N-class Battle. The interaction between them is expressed by an interaction-relationship between the classes Hero and Opponent. The following picture shows the classes and their relationships.
4.4 Example: virtual museum application

As a larger example we consider a typical virtual museum application. The museum consists of sections each of which deals with a specific theme. A section has several adjacent rooms containing the museum's exhibits. There are three kinds of exhibits: paintings, pieces of furniture, and installations (= works of art consisting of sub-objects that move in a complex way). A user can undertake tours through a section of the museum. Such a tour has a specific theme and consists of successive segments. The segments themselves are composed of steps. Tours are guided by a museum guide which walks from room to room and comments on the exhibits therein.

This application is a good example for an HMA that uses all four types of classes at the same time. Therefore, it can easily be shown how the different kinds of classes work together in a common model.

The following picture shows an extract of the domain model for the virtual museum application. To avoid cluttering, attributes, operations and behavior were omitted.

The museum, its sections and rooms as well as the exhibits are spatial objects. Consequently they are modeled with S-classes. Since pieces of furniture, paintings and installations are special exhibits, there is an inheritance-relationship between the corresponding S-classes. The fact that rooms are adjacent and contain the exhib-
its is expressed by an adjacent- resp. a contains-relationship. The museum guide is a
typical agent and is therefore modeled by an A-class. His complex behavior is spec-
ified with a separate N-class (CommentOnExhibit). The tour and its parts are nar-
rative units for which the model contains N-classes. The simultaneity of the steps as
part of a tour segment, and the behavior of the tour guide are taken into account by a
simultaneous-relationship. Participant-in-relationships model which objects partici-
pate in the narrative units. Finally the theme of the tour is specified by a convention-
al class.

5 THE INSTANCES MODEL

The instances model consists of instances of the domain model’s classes, and in-
stantiated relationships. It contains a model component for every object of the run-
ning application. Each model component has a unique name and provides the name
of the object’s class. It is left to the developer whether he specifies the values of an
object’s attributes, as well as operations and behavior differing from the general
specifications given in the class definition.

However, the instances model is not created by a simple and mechanical instan-
tiation of domain model classes. For example, it is possible to aggregate objects into
homogeneous or heterogeneous collections. Thus, the instances model can contain
objects for which a corresponding class can not be found in the domain model.

The instances model is notated graphically like the domain model. Thus, in-
stances are represented by nodes, and relationships by lines between them. A node’s
symbol has a small signet in the upper left corner indicating the object’s kind of
class. The lines for the different kinds of relationships are notated the same way as
in the domain model, i.e. with special arrows or symbols. To avoid huge and confusing
model graphs it is possible to split the instances model into several parts focusing
on groups of objects that belong together.
5.1 Example: virtual museum application (continued)

Continuing the example from subchapter 4.4 we now present the instances model of the virtual museum application. The following picture shows an extract that deals with a segment of a tour through the basement of the Prado museum. This section contains among others the pinturas negras paintings by Goya.

Worth mentioning is the object called PintNegras. Notice that there is no class for it in the domain model. The reason is that this object is a collection of objects of the class Painting.
6 THE REPRESENTATION MODEL

The representation model refers to the domain and instances model and captures aspects of the object representation and the user interaction. It is of vital importance that the representation model does not get overloaded with details that are secondary at this early development stage. Therefore, only the most relevant structural aspects are considered.

Its main model concept are representations. Primarily, a representation models which and how attributes and relationships of an object of the running application are represented to the user. To this end, the media object type (text, graphic, image, audio, video, animation, vrml-world, ...) and a list of output media (window(-part), audio channel, external device, ...) to be used for playback are specified. But a representation still abstains from details such as formats (GIF, JPEG, MPEG, ...) or even concrete names of data files.

In addition it is modeled how several representations build more complex representations. This way it is possible to establish the inner structure of the overall HMA. Besides, this helps mastering the complexity.

After the representations are modeled, they are interconnected via spatial-temporal relationships. These relationships specify where and/or when a representation
is represented in relationship to one or more other representations. These specifications are very coarse on purpose; we just decide whether a representation is represented left, over, simultaneously, after and so forth of another representation.

We consider a navigation as a user-triggered start of playback of an object, executed at another representation. Navigation is specified like spatial-temporal relationships. Depending on whether or not a navigation leads to the termination of the playback of the representation where the navigation was executed, the corresponding line between the two participating representations has one or two arrows in the representation model.

Finally events and user commands are modeled by specifying where they have happened (e.g. in the context of a represented object), what has happened (for example, an audio has reached its end or special object appears in a video) and what the reaction is (e.g. a certain representation changes its size or location). User commands are considered as special events that are executed by the user with the help of certain interaction techniques.

6.1 Example: virtual museum application (continued)

We suppose the virtual museum application lets the user inquire detailed information about a painter by clicking on the pictures of his paintings during the tour. As a result a portrait of the painter and a short text with general information appear in a separate window. At the same time the text is played back as an audio. As soon as the user clicks on certain words within the text, the audio is stopped and a video about the painters life is played back to the right of the portrait. A short while after its end a second video about the influence of the artist follows automatically.

![Figure 7 Window with the details about Goya.](image)

The following picture shows an extract of the corresponding representation model. We assume, that the class Artist has the attributes Portrait, Biography, and Influence. By means of this example, it can be seen how several related representations can be bundled to a complex representation.
7 CONCLUSION

With HyDev we have introduced a new model-based approach that supports the development of hypermedia applications. HyDev defines activities for each development phase. The development process starts with the modeling of the application's objects and their classes. In contrast to similar approaches HyDev works with additional special classes and relationships. The representation aspects (such as media object types, navigation, events, temporal and spatial aspects) are captured in a separate model. These specifications abstain from details and are refined during the specification&design phase.

Benefits of HyDev

Today one can often observe a „just make it“ approach: HMAs are implemented rashly using authoring tools. HyDev however requires an initial determination and
modeling of requirements. This way a developer is caused to consider and examine the application more thoroughly, resulting in a better understanding of the system to be built. Problems are identified early, and new ideas can evolve. As a consequence, the end product will be of higher quality concerning correctness, robustness and maintainability.

The various models are a mandatory basis for the activities of later development phases. In this respect, authors, designers, developers, and programmers are given a starting point for their work. Besides, the information captured in the models serve as an extensive documentation, especially for maintenance purposes (modifications, extensions, elimination of errors). Up to now, developers usually apply fairly informal documentation techniques like storyboards.

Certainly, the making of the various models costs time. But this extra effort pays off when it comes to implementation or maintenance of an existing product. Apart from that, a developer must think about these issues anyway. Currently, he/she does so on the side and disorderly, at worst during implementation. In any case it is better to analyze the intended application and capture the findings early and systematically.

**Future work**

So far, we have analyzed several existing products of various categories. **HyDev** was developed based on the resulting observations and findings. Currently, we are refining some aspects of the representation model. Furthermore, we are engaged in additional case studies, with the aim of validating our approach.

In the future, we will consider the activities of the specification&design and the implementation phase in more detail and develop appropriate models. In addition, we will provide a sophisticated edit tool offering extensive support for the creation and modification of the various **HyDev** models. Among the features of this tool will be elaborate drag&drop functionality, consistency checking, semantic zooming (the more we zoom in the more details can be seen), and selective viewing (e.g. only the directly connected neighbors of a selected object are shown; only objects of a specific class are shown). For that purpose we will adapt the generic edit tool **GenTool** which was developed in the context of our work on the FLUID-approach (Homrighausen et al., 1997; Kösters et al., 1996).

Finally, we will pay special attention to the issue of rapid prototyping. The rapid development of prototypes is especially advisable and realistic in the hypermedia domain. Since due to their complexity and abstractness the models are not suitable to be directly used for discussions with customers and/or users, we will examine how prototypes of HMAs can be generated (semi-)automatically on the basis of **HyDev** models.
8 REFERENCES


9 BIOGRAPHY

Peter Pauen is a research assistant at FernUniversität Hagen. His research interests include hypermedia applications and software engineering.

Josef Voss is a research assistant at FernUniversität Hagen. He received a Ph.D. in computer science in 1990. His research interests include user interface tools and user interfaces in requirements engineering.

Hans-Werner Six is a full professor in computer science at FernUniversität Hagen since 1985. He received a Ph. D. from University Karlsruhe in 1978. His current research interests include Geographical Information Systems, Software Engineering, and graphical user interfaces.
Design Support for Hypermedia Documents

M.D. Apperley and R.B. Hunt
University of Waikato
Hamilton, New Zealand, m.apperley@cs.waikato.ac.nz

Abstract
The nonlinear nature of hypermedia documents makes them notoriously difficult to describe. Consequently design prior to implementation is a challenging task. This paper examines design issues specific to hypermedia, and describes the development of HANDIE, a notation for the description of the structural organisation of hypermedia documents. HANDIE is based on a directed graph approach, but it incorporates a range of abstractions which provide significant simplification, and which allow the underlying structure of a document to remain clearly visible. The evaluation of a prototype design environment based on HANDIE is described, and range of refinements for the future are proposed.

Keywords
hypermedia, multimedia, design methodology, World-Wide Web, graphical notation
1 INTRODUCTION

Because of their inherent nonlinearity, hypermedia documents are very difficult to describe; in fact, their best description is usually their implementation. As a consequence, attempting to carry out comprehensive organisational and structural design prior to implementation is a challenging task. The majority of hypermedia design aids available are either (i) style guides, relating to appearance and frame-to-frame relationships, providing little to support the overall design of the document or system, or (ii) are inextricably tied into a particular formal design methodology. The development and extraordinarily rapid penetration of the World-Wide Web has highlighted the need for tools to assist with WWW site design and maintenance, where the bulk of the issues involved are generic to hypermedia. This need is further exacerbated by the fact that many of those engaged in the design of hypermedia systems and WWW sites have little or no formal background in software systems design (Pohl and Purgathofer, 1994).

This paper examines the critical problems of hypermedia design which set it apart from more conventional software development, and reviews a range of current design tools. From this base, a graphical notation for the description of hypermedia systems is then developed, aimed specifically at supporting the hypermedia design process rather than prescribing a formal design methodology. The prototype implementation of this notation within a hypermedia design environment (HANDIE) is described. Some preliminary feedback from designers who have evaluated the system is also discussed.

2 HYPERMEDIA DESIGN ISSUES

Designing hypermedia documents is a complex activity. Nanard and Nanard (1995) identify the need for formal tools to reduce the cognitive load on the document designer, and to reduce the level of complexity in the design process. One approach is to provide the user with abstract semantic types which can be used to describe a complex real-world situation in a relatively simple fashion. For example, Entity-Relationship (ER) modelling is used in traditional software design with the provision of abstract types in the form of entities and relationships at the simplest level, followed by more complex semantic constructs such as aggregation and specialisation at higher levels (Chen, 1976). Another approach to reducing complexity is to allow the user to trial portions of the design in an incremental, experimental and evolutionary fashion. This is usually achieved through rapid prototyping where the user is presented with a working prototype of the user interface without the underlying functionality.

This need for formal tools and complexity reduction is not unique to hypermedia design, but rather applies to the design of any software system, and as a consequence the topic has already received considerable attention, with varying
success. However, as Nanard and Nanard (1995) have pointed out ‘An important part of hypertext design concerns aesthetic and cognitive aspects that software engineering environments do not support.’

Hypermedia document design is most commonly carried out or described at the level of the data, rather than at a higher metadata or semantic entity level. Because of the inherent network nature of hypermedia documents, it is tempting to use standard directed graphs for their description. Figure 1 shows a directed graph representation of part of a university WWW site providing information on lecturers and courses. A standard hierarchical structure provides links from an introductory page to an index of lecturers and an index of courses, and from these indexes to individual lecturers and individual courses respectively. However, links are also provided from lecturers to the courses which they teach, and vice versa, and from each node back to the introduction. For just a few lecturers and a few courses, Figure 1 demonstrates the complexity of this directed graph representation. Further, because of the visual complexity of this representation, Figure 1 also shows the apparent opacity of the underlying structure that it represents.

![Directed Graph Representation](image)

**Figure 1** A directed graph representation of a simple hypermedia document, showing the complexity and semantic opacity of representation at this level.
A further obfuscating aspect of hypermedia documents is that there are potentially three overlapping and conflicting structures, some or all of which may be explicitly represented. These are (i) the logical, inherent or epistemological structure of the information, (ii) the imposed or navigational structure, and (iii) the user's perceived structure (Jul and Furnas, 1997). It is usually only the imposed structure (ii) that is represented in diagrams of the form of Figure 1.

3 EXISTING DESIGN TOOLS AND METHODOLOGIES

There are a number of existing tools and procedures that support the hypermedia design process. These variously encompass simple descriptive techniques for the actual structures at the data level, abstraction techniques to allow modelling and description of the conceptual structures, and formal methodologies aimed at improving the design process. The Object Oriented Hypermedia Design Model (OOHDM) (Schwabe and Rossi, 1995; Schwabe et al., 1996) describes hypermedia development as a four-stage process, comprising conceptual design, navigational design, abstract interface design, and implementation, in that sequence. It focuses on navigational and abstract interface design, and encourages re-use. The Relationship Management Methodology (RMM) (Isakowitz et al., 1995) echoes the phases of OOHDM, but adds an additional step between conceptual design and navigational design, that of slice design. Slice design determines how the information defined in the entities (the attributes) of the conceptual model will be grouped and presented to the user, and provides a high-level partitioning of the logical and imposed structures. For example, for the system of Figure 1 it might be decided to store the following information about each lecturer entity: name, photograph, biography, research area. The slice diagram of Figure 2 shows exactly how this information would be structured in the document, which attributes would be associated with each slice, and how this information could be navigated.

![Figure 2](An RMM entity slice diagram.)
The Dexter Hypertext Reference Model (Halasz and Schwartz, 1994) presents a similar approach, and could be regarded as the predecessor of OOHDM and RMM. All three of these methodologies support abstraction and provide an enabling descriptive notation. The Hypertext Analysis, Navigation and Design model (HAND) (Duncan and Apperley, 1994) is simply a visual notation for high-level representation of hypermedia structure, without an accompanying methodology. However, all of these approaches share the common feature that they provide abstract semantic types for specifying structure and navigation, in contrast to earlier design approaches such as that of HyTime, which merely provided mark-up languages for specifying the links between different parts of the document (Newcomb et al., 1991). The semantic types provide by these models range from the simple representation of nodes and links in the Dexter model, to guided tours and entity indexes in RMM, and to classes and groups in HAND.

There are also differences between these models in how the node and link concepts are represented. Some, for example RMM and HAND, regard the node as the atomic unit, and all links are specified as emanating from some source node and terminating at some destination node. Others, notably OOHDM and the Dexter Model, store information about anchors* within node definitions, and thus allow finer-grained link specification, from a source anchor to a destination node. Some models also make a distinction between atomic and composite nodes. Atomic nodes contain a single type of hypermedia, such as a block of text or a video clip, while composite nodes can combine a number of media types in a single node (Duncan and Apperley, 1994).

The discussion so far has dealt specifically with hypermedia. Hypermedia documents increasingly incorporate elements of multimedia as well as hypertext. With applications which include video or sound data, some consideration must be given to timing constraints. The Amsterdam hypermedia model (Hardman et al., 1994) combines multimedia and timing considerations with the abstract node and link representations of the Dexter hypertext reference model. More recently IMMPS (Shih and Davis, 1997) has provided a development environment for multimedia presentations incorporating AI techniques for specifying knowledge inheritance, and a multimedia database.

4 THE DEVELOPMENT OF A GRAPHICAL NOTATION FOR HYPERMEDIA DOCUMENTS

The HANDIE hypermedia notation described in this paper, which is based on an extension of the earlier HAND notation (Duncan and Apperley, 1994) has been designed specifically to address those design issues unique and problematic to hypermedia document designers. It makes no attempt to reproduce or replace

---

* See Ginige et al. (1995) for a useful set of definitions of such terms
methodologies such as RMM or OOHDM, but rather, it aims to provide ‘...concepts and tools that help produce a design and (sometimes) implement the corresponding product’ (Nanard and Nanard, 1995). To this end it must both provide adequate formalisms and at the same time support the incremental and opportunistic activity of the designer (Nanard and Nanard, 1995). These requirements have led to the following four general principles which were adopted in the development of HANDlE:

- **Transparency:** The notation must be easily understood by users, who may not have formal training in software development.
- **Completeness:** The model must provide a sufficient range of abstract semantic types to aid the user in reducing the complexity of the hypermedia design process.
- **Software Implementability:** The model must be able to be supported by a computer-based development environment.
- **Continuity of Support:** The model should provide for the graceful transition from early design right through to implementation, or at least provide a continuous link in this path.

The HAND notation (Duncan and Apperley, 1994), which provided the basis for HANDlE, already meets a number of these requirements. It concentrates on the structural and navigational phase of design through the use of directed graphs, and incorporates higher level concepts which both simplify the representation of the underlying structure and at the same time make this structure more evident.

The original HAND notation provides four abstract node types, illustrated in Figure 3 (a to d). These node types can be described as follows:

- **Basic nodes:** These are nodes that represent a single document page. A basic node may contain a single media form, or it may be a composite of several media.
- **Group nodes:** Group nodes provide a means of representing hierarchy, and simplifying views. A group node corresponds to a section of the document which is represented by a sub-diagram; it can be thought of as ‘containing’ a collection of nodes and links. A group node is a similar concept to that of an HM s-collection (Maurer et al., 1995) or an RMM slice (Isakowitz et al., 1995).
- **Class nodes:** A class node represents an homogenous group of related node instances, each with the same underlying structure. Each node instance in the class can be thought of as being similar to a record in a database.
- **External nodes:** These provide a mechanism for developers to establish links to and from nodes outside the scope of the current document.
In HANDIE there is one further node type, the list node, illustrated in Figure 3(e):

- List nodes: Class nodes represent homogenous groups of related node instances. A list node provides a means of accessing individual instances of a class node via an index.

![Diagram of abstract node types](image)

**Figure 3** The five abstract node types of HANDIE, and their representations: (a) The basic node, (b) group node, (c) class node, (d) external node, and (e) a list node.

The HAND notation included two forms of link, simple and complex (Apperley and Duncan, 1994). A simple link represents a direct relationship between two nodes, with a fixed anchor and a fixed destination node. A complex link, however, is a link for which the destination is determined at runtime, on the basis of contextual information. HAND provided no means for specifying this context. HANDIE has expanded and refined the notion of a complex link, to that of a conditional link. A conditional link will always have as its destination a class node, and includes a specification of the condition associated with that link which will determine the actual destination instance at runtime. Examples of conditional links can be seen in Figure 4, where such links are represented by thick lines and are accompanied by associated query specifications.

A further concept that has been included in HANDIE is that of the higraph (Harel, 1988). Higraphs provide for further complexity reduction in directed graph representations by allowing nodes to be grouped into sets; a link specified from a node set to another node is equivalent to a link from every node in that set. The higraph concept as implemented in HANDIE is illustrated in Figure 5. A comparison of the HANDIE representation of Figure 4 and the conventional directed graph of Figure 1, demonstrates how the use of higraphs and class nodes significantly reduces the number of links without compromising the explicitness of a diagram.

To illustrate the use of the group node in HANDIE, consider the situation (from Figure 4) where the lecturer for one course wishes to establish a collection of web pages to support that course. There should be a link to that collection from the course information page, but the detail of this collection is not of part of the design at present. The appropriate representation of this situation, using a group node, is shown in Figure 6. Note here a further use of the conditional link notation. A
conditional link is used in Figure 6 with a class node (Courses) as its source; in this case it indicates which instance (or set of instances) of the class is the anchor for what is, in effect, a simple link.

Figure 4 An example document design using HANDIE.

Figure 5 (a) Standard and (b) higraph enhanced multiple link abstractions.

Figure 6 also provides an illustration of the use of the external node abstraction. This provides a mechanism for a link from the University's home page to the
introducory node; the home page will always remain outside the scope of the
design, yet it is necessary to make reference to it.

HANDIE, as developed at this stage, is concerned principally with the definition
of the navigable structure of hypermedia documents, and does not provide full data
modelling facilities for specifying the data contained in those documents. Ultimately it is anticipated that modelling tools will be integrated into the
HANDIE environment. However, for the present, limited provision has been
provided for the specification of existing data models. For each entity, the user is
able to specify the relevant attributes. Sets of these entity-attributes can then be
associated with individual nodes, and are then available for use in conditions
associated with links between nodes. The tables of Figure 7 show the attributes
associated with the lecturer and course entities of Figure 6, and the data contents
of the index and class nodes. These tables help explain the link conditions in
Figure 6.

**Figure 6** An illustration of the use of the HANDIE group and external node
facilities.
Figure 7  (a) The data model used in the design of Figure 6, and (b) the data contents of individual nodes.

5  THE IMPLEMENTATION OF A PROTOTYPE DESIGN ENVIRONMENT

The HANDIE hypermedia design tool described in the previous section has been implemented as a stand-alone Java application using JDK 1.1.3 in a Windows NT environment. The basic design window of this prototype system is shown in Figure 8. Apart from the appearance of handles on nodes and links, the representation of higraphs, and the display of link conditions, there is little difference between the appearance of the actual design of Figure 8 and that of the idealised form of Figure 6.

Within the HANDIE design window the user is able to specify nodes of specific types from a menu and to drag them to the desired position using the mouse, and to define links by dragging from a handle on the source node to a handle on the destination node. Individual nodes and links can be deleted (via the Node and Link menus) and nodes retain all link connections if they are moved by dragging. The higraph link from the four nodes Lecturer Index, Course Index, Lecturers and Courses is represented by the short sourceless link to Introduction in Figure 8. This is produced by explicitly drawing the four separate links, selecting them as a group, and then selecting Higraph from the Link menu. Clicking on this abbreviated higraph link on the design causes the component links to be displayed in highlighted form.
The data model for the design is defined in a separate pop-up window which is accessed through the Data menu in the design window. An example of this data definition window is shown in Figure 9(a). Once this information has been provided, then the data contents for a specific node can be defined and accessed by a double-click on the node (pop-up window of Figure 9b), and conditions associated with a given link can be defined and accessed by a double-click on the link (pop-up window of Figure 9c). Note that for reasons of clarity, link conditions are not permanently displayed as suggested in the idealised form of Figure 6.

The implementation includes full support for group nodes and hierarchical designs. A double-click on a group node causes a new design window to be opened, within which the lower-level sub-design may be carried out. Through judicious use of this feature, users are able to maintain their designs at a manageable level of complexity and avoid performance limitations that may arise through designs with large numbers of nodes.
Figure 9 (a) The data definition pop-up window in which relevant data entities and their attributes are defined, (b) the pop-up window for defining the entity-attributes associated with a particular node, and (c) the definition of the condition associated with a link.

6 PRELIMINARY EVALUATION BY DESIGNERS

A preliminary evaluation of this prototype tool has been carried out by five users whose work involves hypermedia document design, computer graphic design or WWW page design. In its present prototype form, performance (interactivity) is satisfactory for designs involving no more than about twelve nodes. Users were given a brief introduction to the tool, and then asked to apply it to a design on which they were currently working. After a half-hour session they were asked a series of questions about the usefulness of the HANDIE notation and the tool as implemented. In general, these users found that the notation supported a wide range of hypermedia constructs, and that it was very useful in conceptualising the structure of a document without focusing on its content. In particular, they found the class and list nodes to be very useful abstractions. However, they also found the notation limiting, with two constructs in particular noted as absent:
• There is no provision for representing a node containing more than one list.
• Class nodes of lists, which would allow lists to point to lists (hierarchical menus) are not supported.

These designers also felt that although the group node abstraction provided a useful partitioning of a design, inevitably designers would want additional links between groups, other than through the head node. It was further noted that none of the designers who used the system made use of the higraph facility.

Differences between users from a programming background and users from a graphic design background became apparent during the evaluation. Concepts in HANDLE such as class and list nodes, and the approach to data definition, required little explanation to the programmers, but were seen to be significant hurdles for graphic designers.

7 CONCLUSION

This paper has described the development of HANDLE, a notation for the support of hypermedia document design. A prototype design environment based on this notation has been developed, and a preliminary evaluation carried out by several people involved in hypermedia document design. This experience both reinforces the idea of the general approach adopted, that of providing tools to support design rather than attempting to impose rigid methodologies, and suggests that there is definite merit in the simple yet flexible notation of HANDLE.

From the evaluation, it is possible to review HANDLE with respect to the four general principles mentioned earlier:

• **Transparency:** Nonprogrammers did experience some difficulties with concepts fundamental to HANDLE.
• **Completeness:** HANDLE was found to deal with most situations, but some additional abstractions were discovered to be desirable.
• **Software Implementability:** HANDLE has already been implemented as a computer-based design environment.
• **Continuity of Support:** HANDLE supports only part of the design process, but it is not inconsistent with what comes before and what follows on.

Obviously some further refinement of HANDLE should be carried out to bring it closer to these principles. Within the scope of the current prototype, the following refinements are currently being considered:

• A composite node facility should be developed, which will allow a single node to be composed of a number of elements, thus allowing for more than
one list in a single node, or a node to combine a list with other information, for example.

- The list node concept should be generalised to allow conditional links from a list node to nodes other than class nodes, allowing lists of lists, or even lists of basic nodes.
- A mechanism should be provided to allow links from one design window to another, so allowing more flexibility in the use of group nodes.
- Consideration should be given to improving the methods for defining and representing higraph links.
- The interim mechanism for data definition should be reviewed.

Ultimately it is intended that HANDIE should be integrated with an appropriate data definition tool on one side, and move closer towards the generation of actual documents on the other. However, a production version of its present form would nevertheless provide valuable and usable support for the overall design process.

8 REFERENCES


**BIOGRAPHY**

Mark Apperley is professor and chairperson of the Department of Computer Science at the University of Waikato, Hamilton, New Zealand. He has been actively involved in human-computer interaction research for the past twenty-five years. His current interests include information visualisation and presentation, computer-supported collaboration, and graphical design notations.
Developing Multimedia Principles from Design Features

F. Nemetz  
P. Johnson

Department of Computer Science  
Queen Mary and Westfield College  
University of London

Mile End Road, London, E1 4NS, UK  
Tel: +44 (0)171 975 5224  
Fax: +44 (0)181 980 6533  
{fabio, petej}@dcs.qmw.ac.uk  
URL: www.dcs.qmw.ac.uk

Abstract

The rapid growth of multimedia technology has made it possible to deliver high quality audio, graphics, video and animation to the user. However, this growth in technology has not been met by a growth in design knowledge. While it is possible to have multimedia it is not at all obvious that we know how to design high-quality multimedia systems that are fully usable to the degree we should expect. To improve the situation much work is under way to develop guidelines, style guides and principles for multimedia design. In this paper, we consider what areas might be in need for investigation in order to derive design principles. Examples of these areas are given and a research agenda for developing principles for multimedia systems is offered.

Keywords

Multimedia system design, principles for multimedia design, multimedia features
INTRODUCTION

Multimedia is a technological achievement that currently lacks a theoretical basis for reasoning about its utility and effects on usability. Using the most advanced technology will not necessarily improve the usability of current designs. Relying upon naive assumptions, beliefs and intuitions alone will not be enough to bring about a widespread improvement in the quality and usability of interactive systems through the use of multimedia. Although multimedia technology can increase the options open to the user-interface designer (Alty, 1997), it has not yet been met by a growth in design criteria and knowledge.

A common view of multimedia is that it is simply the use of more than one medium to present information to users. We adopt a wider definition; encompassing both input and output media and focusing on human-computer interaction rather than on the technological aspects. In this way, we consider interactions with animations, gesture recognition, speech input, speech synthesis, haptic input and output, hypermedia and virtual reality as pertaining to multimedia. As Marmolin (1992) states:

'A user centred definition would characterise multimedia systems as systems enabling the usage of multiple sensory modalities and multiple channels of the same or different modality (for example both ears, both hands etc.), and as systems enabling one user to perform several tasks at the same time. That is, multimedia is viewed as a multisensory, multichannel, and multitasking approach to system design. In addition multimedia systems put the user in control, i.e. could be described as a user centred approach' (Marmolin, 1992).

Traditional approaches to design for usability from Human-Computer Interaction do not yet directly deal with the unique characteristics of multimedia systems: 'while general usability criteria such as learnability, flexibility and robustness apply equally to single media and multimedia systems, they have little to say regarding the specific benefits and drawbacks of concurrent media input and output' (Bearne et al., 1994).

The use of multiple media, when well exploited by designers, potentially makes multimedia interfaces more exciting, more natural, more enjoyable and pleasant to use than traditional mainly text-based interfaces (Petersen, 1996). This occurs because multimedia provides us with richer forms of representing information in human-computer interactions. However, it does not necessarily follow that merely by increasing the richness of the media we will increase the utility and usability of computers and the information. While in some cases the addition of more media will allow us to express concepts and information more fully, with greater clarity,
and with greater accuracy than before, in other cases it will introduce ambiguity, confusion and contradiction.

Our research aims to define a set of principles to address the complexities of multimedia design and evaluation, in order to make multimedia systems useful and usable, rather than ‘gimmicky’ and ephemeral. The principles that will emerge from this research are expected to support designers in making decisions about the various media so as to maximise the effectiveness and efficiency of the user-computer interactions. This will enable designers to build more usable multimedia systems, moving from a craft style design approach to a more systematic principled-based approach.

In the following section, we present some of the design issues we are focusing our research upon, and for which we hope to be developing the principles. We include some examples to illustrate them. Finally, we discuss the basic steps involved in the continuation of this research.

2 TOWARDS MULTIMEDIA PRINCIPLES

The term principle is being used in different ways in the literature. Shneiderman (1997) differentiates between three kinds of guidance for designers: high-level theories and models, which offer a framework or language to discuss issues that are application independent, middle-level principles, which are useful in creating and comparing design alternatives, and specific and practical guidelines, which provide reminders of rules uncovered by designers. For instance, one of his middle-level principles is ‘Use the Eight Golden Rules of Interface Design’ which includes eight design recommendations (e.g. enable frequent users to use shortcuts). This agrees with his statement that ‘the separation between basic principles and more informal guidelines is not a sharp line’. Yet, Preece et al. (1994) consider principles to be a special case of guidelines. For them, there are two kinds of guidelines: high-level guiding principles and low-level detailed rules. They consider principles as guidelines that offer high-level advice that can be applied widely (e.g. know the user population). On the other hand, principles and rules are considered to be synonyms by Baecker et al. (1995): ‘collections of statements that advise the designer on how to proceed (e.g. know the user)’, while guidelines are defined as ‘collections of tests that can be applied to an interface to determine if it is satisfactory (e.g. provide an average response time of less than one second)’.

We consider a principle to be some established fact that has a theoretical and empirical basis for its acceptance, that can be applied to a prescribed problem area in a well-defined manner and for which there is some indication of what the result of following the principle (or not) will be. At this stage in our research we are able
to point to areas where we believe we ought to be developing a more principled understanding of interacting with multimedia and what might be the features that an underlying set of principles for multimedia design have to address. The areas presented here resulted from a literature survey, the main source being Alty (1993). They seem to help to understand and explain the complexities of multimedia systems, and are serving as the basis for us to pursue the search for the aforementioned principles. Some of the features are particular to multimedia, while others are more general to wider areas of HCI.

2.1 Naturalness and Realness

Multimedia systems try to take advantage of human senses to facilitate human-computer communication, and human-human communication. Considering that we live in a world of multimedia events (Rudnicky, 1992), ‘many people believe that multimedia communication is natural and corresponds more closely with how the brain has developed’ (Alty, 1997), and, therefore, multimedia exercises the whole mind (Marmolin, 1992). In this viewpoint, the human brain is seen as having evolved in a multisensory environment, where simultaneous input on different channels was essential for survival. Thus, ‘the processing of the human brain has been fine-tuned to allow simultaneous sampling and comparison between different channels’ (Alty, 1997). Multimedia systems have the potential to make more appropriate and efficient use of human perceptual and cognitive capabilities, by making the interaction more natural. In this sense, a better understanding of how our perception and cognition are affected by a particular medium and by their combination is needed.

A related feature to naturalness is realness, or the degree of correspondence to the real thing. Naturalness and realness are similar but not the same. Naturalness here is concerned with the mapping between the stimuli and the senses, taking recognition of the fact that people normally gain information about the world from multiple senses (e.g. hearing an explosion would cause people to look for a cloud of smoke). On the other hand, realness is concerned with how close the representation of the explosion corresponds to the actual explosion.

Two consequences for systems that possess these features appear to be that they show properties of believability (the closer to the reality, the more believable) and fidelity (degree of detail and faithfulness). Hence, in figure 1 the representation of the document has a high degree of realness (i.e. it closely corresponds to the appearance of the actual document) and naturalness (i.e. it is perceived through our visual object recognition system).
2.2 Media Allocation

How, and on what basis, is a particular medium selected for the presentation of a particular piece of information? Each medium has both constraining and enabling features (Arens et al., 1993), affords different interactions, offers different communicative intentions and has its own rules and conventions.

But is it enough to have a knowledge of each medium in order to make an adequate selection? Some argue that it also depends on the user’s knowledge and experience of a domain and task: if the domain and task are new to the user, a concrete representation that allows exploration seems to be best; if the user has a lot of experience in the domain and task, more abstract representation may be adequate (adapted from Marmolin, 1992). Alty (1993) adds that the usefulness of different media in presentation situations is closely related to the complexity of the idea being conveyed. Nevertheless, he also states that the capabilities of the perceiver play an important role on the media allocation problem.

There is an important difference between abstract and concrete concepts. Abstract and complex concepts are more easily and completely represented by words than by pictures. In contrast, more concrete concepts, if represented by pictures and sounds, can improve the speed of understanding and comprehension over that of text representation. Moreover, the choice of medium also has to consider what information is intended to be conveyed and what is the intended effect of the information.
It is not easy to define a complete set of criteria to solve the media allocation problem. One aspect that should be investigated in detail is the relation between media and tasks. In other words, the main problem is to establish which media best transmit the information needed by the users to carry out their tasks.

Summarising, it seems that multiple factors play a role in the media allocation decision (Arens et al., 1993):

- Characteristics of the media
- Characteristics of the information
- Goals and characteristics of the user
- Goals of the producer.

Based on these factors, it is necessary to determine the enabling and constraining features of each medium, given the goals and characteristics of the user, the intended effect of the information and the characteristics of the information itself. Then, it will be possible to determine the media to be used.

In a CD-ROM produced for orthopaedists (Evolucao, 1996), there were several possible ways to show the manoeuvres employed to make a diagnosis about a given joint problem. In books, they are usually presented by pictures or by abstract sketches. In the particular CD-ROM, video (figure 2) with audio and text explanations were used to show the dynamics and clarify the important aspects of the particular manoeuvres in a way that the media could match to the nature of the information, the goals and skills of the users and the purpose of training.

![Figure 2: A video frame from the CD-ROM 'Semiology of the Knee' (Evolucao, 1996).](image)

### 2.3 Redundancy

Often considered useful in complex and cognitively laden tasks, redundancy is considered a significant phenomenon in multimedia systems (Vetere, 1997). It is
well known that using both visual and audio channels simultaneously to explain a complex diagram can be better than using only one channel; it is also true that 'human beings use the redundancy offered by multiple channels to improve their understanding of situations' (Alty, 1991). Redundancy is related to naturalness, when we consider the multiple sensory input channels a person uses. However, redundancy relates to the information content of stimuli rather than their forms.

In multimedia systems, redundancy is achieved through the integration and synchronisation of different media. It can produce 'real-world' like conditions, and reduce the overload on working memory (e.g. video and audio, animated graphics and text overlay (or sound commentary)). Comprehension is directly affected by redundancy, since there is more chance of the information provided being understood. For instance, if there is confusion and misunderstanding as a result of a misperception of information in one medium, then this can be supplemented by providing the same material in another medium, at the same time (or proximal in time).

Understanding how to use redundancy effectively is still a challenge for multimedia systems designers. If combined in a congruent (harmonic, synchronised) way, the use of multiple media are far more effective than the using a single medium (Hoogeveen, 1997). However, if combined in a non-congruent way, they are less effective (in this case, disruption, ambiguity and confusion occur). Multiple-resource theorists have argued (Anderson, 1995) that human beings have multiple resources and that how much two tasks interfere with each other depends on whether they make demands for the same resources. Paivio (1986) in his dual-coding theory states that we have two separate but interdependent information processing systems: a verbal system (specialised for dealing with linguistic information) and a visual system (specialised for processing non-verbal objects). In experiments with multimedia learning, Mayer (1997) showed that, if the verbal and visual modes are coordinated (e.g. words with pictures, animation with narration), it is possible to produce significant improvements in understanding and learning. It helps learners to select visual and verbal information and to build one-to-one connections between actions in the visual representation and in the verbal representation.

Vetere (1997) states that presently there is insufficient knowledge to help designers manipulate these redundancies to improve interactions. No methodology or criteria on how to apply redundancy in multimedia systems have been developed so far, let alone a theory of redundancy and its effects on usability.

To exemplify the use of redundancy, in figure 3 a document is presented in two ways: as a photographic reproduction and as a textual transcription. Even though the photograph can be zoomed in, the transcription is easier to read and to search
for. This is an example of redundancy, where two representations of the same information are presented in different formats.

Figure 3: A document and its transcription (Corbis, 1995).

2.4 Significant Contribution of the Media

The opposite of redundancy is information richness. However, this can lead to information overload. Just adding a new medium will not guarantee an improvement in the user’s ability to recognise and understand how to interact with a given system or the meaning of a particular piece of information; it can even exceed human attentional capabilities for handling multiple sources of information (Bearne et al., 1994, Barnard & May, 1995). Hence, additional media should be used only if they make a significant and relevant contribution in the transmission of a message. Otherwise, it can distract the user, making her lose attention to what is required.

It is important to observe that this kind of problem also occurs in everyday general communications. Our understanding of multimedia can greatly benefit from many communication theories. Grice’s theory of implicature (Levinson, 1983), for instance, is concerned with the efficient and effective use of language in conversation. One of its maxims, the maxim of quantity, is related to the fact that, when making a contribution to a conversation, this contribution must carry all and only the necessary information, not more and not less than what is required. Another maxim, the maxim of relevance, states that one should make his/her contributions relevant.

In a hypermedia application - a literature multimedia encyclopaedia (Nemetz et al., 1996) - we can find an example of this feature. Figure 4 shows a passage of a book that illustrates a particular characteristic of an author. This passage is presented in text and, optionally, in audio. The audio is actually composed by two channels: the first contains the reading of the passage by a narrator, and the second contains an audio-effect that resembles the sound of wind. This effect provides an
atmosphere to the narration, associating its contents with the name of the book: *Time and Wind*. In this case, the audio-effect makes a fundamental contribution, because it is actually reinforcing this association, enhancing its semantics and making it more pleasant, but at the same time it does not seem to hinder comprehension.

Figure 4: A passage from a book (Nemetz et al., 1996).

2.5 Exploration

One of the main advantages of multimedia systems seems to be the increased level of interactivity* they provide. This happens not only due to the use of our senses in a fuller and more orchestrated manner, but also because of a greater flexibility and freedom to explore the information. Ideally, neither the author nor the designer should decide how the information should be processed (Marmolin, 1992); the user should be in control, exploring the interface and choosing the best media for the task.

Exploration is a desirable property of general HCI in that it allows users to discover the workings, content and functional use of a system (Carroll, 1990). Multimedia can be used to facilitate greater exploration in all these areas, but the media have to be designed to support exploration. The ability to support the user in exploration itself has to be designed; it does not just happen.

---

* Level of interactivity, in this context, is the degree to which a computer system is responsive to the user's (explorative) behaviour (Hoogeveen, 1997)
A high level of interactivity improves sensory stimulation, and thus facilitates human information processing (Hoogeveen, 1997). Alty (1993) adds that, for ill-defined goals (or goals not well understood), it is better to allow users to exploit the interface and choose the best media for the task. And Bearne et al. (1994) suggest, in their usability guidelines for multimedia systems, that users must be given control over the appearance and the disappearance of each piece of information. The feeling of engagement produced by the freedom of exploration is an important issue to take into consideration when designing a multimedia system.

One possible explanation for this phenomenon is that we explore our environments in an active way; we are not passive receivers of information (Marmolin, 1992). Quoting Gibson (cited by Marmolin (1992)), 'we do not hear, we listen; we do not see, we look around'. This is consistent with the principle in active learning that users learn best when they are actively involved and creating. The system should explicitly afford exploration, inviting the user to explore it, providing appropriate feedback to each action, and offering an easy way to reverse any action, thus providing a safe environment for exploration.

An example of the exploration feature is the slide show facility (figure 5) provided in (Corbis, 1995). With this tool, the user can prepare customised guided tours of paintings based on his own criteria. Although there are several guided tours available, giving this possibility to the user, allows him to explore the system in a more active way according to his goals, rather than being just a passive viewer. The only problem is that, in this particular case, the user is not able to add annotations or audio to the presentations, which would give him a better way to actively explore the contents of the system.
2.6 Quality of Information Representation

It has been argued (Hoogeveen, 1997) that the quality of the representation of multimedia information (e.g. graphic representation x photographic representation) can affect the way people interact with multimedia systems. Each medium has its own rules and conventions and will make its own special demands and requirements upon technology to enable that medium to be used optimally. Although literacy is required in every medium, most software designers are not well skilled in film or video presentation languages (Alty, 1991). People are used to high-quality productions (such as in films or television) and could expect to see something of the same standard in a computer display.

With today's technology, current multimedia representations often have poor quality if compared with their analogue counterparts. For instance, a digital video in a small window cannot compete with the quality provided by an ordinary television. Even though for some kinds of applications this quality is enough (e.g. video-conferencing), for others it can be a restrictive factor (e.g. remote diagnosis by a dermatologist). Therefore, we still do not have a full realisation of the potential of multimedia, although the adequate quality depends ultimately on the task the user is performing.

4 DISCUSSION AND CONCLUSION

The features presented in this paper reflect many of the main aspects of multimedia systems. At the present stage, they can be considered as design problems that would require principles to guide their solution. It is important to note that, although some of the features can be general, i.e. not specifically addressed to
multimedia systems (e.g. redundancy, quality of information representation, exploration), they do represent pertinent aspects of multimedia systems design. It should also be noted that the desired principles may not be equally applicable to all classes of multimedia systems and domains.

The features presented here are a step above guidelines. The problem to be addressed is to formulate principles in such a way that:

1. Each principle must embrace at least one feature, and the set of principles should embrace all the features.
2. The principles have to be somehow generically applicable, but at the same time detailed enough to be tested.

In order to advance the research, first we need to refine these features into a set of principles, so that they can be expressed in a more complete and systematic manner, including examples, appropriate theoretical and empirical evidence, and to make predictions about their effects on usability.

The next step is to assess and refine the principles on different classes of multimedia systems, domains and tasks. In doing this, we will assess their (i) predictiveness and reliability through experimental testing, and (ii) applicability and usability through use in design context. In this way we will be assessing if they apply to multimedia design problems, if they can predict usability issues and be applied to those issues, and if the principles themselves are usable by designers and evaluators to develop and evaluate the usability of systems using multimedia.

In the end, we should be able to propose evaluation and design methods or techniques that are principle-based. A method of evaluation would include criteria, data, analysis and interpretation to produce redesign recommendations. In order to support design creation, an environment could be developed, which would include exemplars, guidelines and constraints derived from the principles.

In this paper, we showed that we need a principle-based approach for the design and evaluation of multimedia systems. We proposed a tentative set of six features that were elaborated with evidences from the literature. The features are:

1. naturalness and realness
2. media allocation
3. redundancy
4. significant contribution of the media
5. exploration
6. quality of information representation.
This is an on-going research topic. In order to achieve our goals, these features need to be further refined, tested and used in real-world situations before they can emerge as principles for multimedia design.

Our research aims to develop basic principles for the design and evaluation of multimedia systems. We believe that these principles will provide a consistent basis for user-interface designers to make better decisions and, hence, to build more usable and useful multimedia systems.

6 REFERENCES


Petersen, M. G. (1996) Evaluating Usability of Multimedia Interfaces. HCI MSc Project Report, Department of Computer Science, Queen Mary and Westfield College.


7  BIOGRAPHY

Fabio Nemetz holds BSc and MSc in Computer Science. Currently, he is a PhD student of Computer Science at Queen Mary and Westfield College, University of London, funded by CAPES/Brazil. His interests are HCI and design and evaluation of multimedia systems.

Peter Johnson is Professor of Human Computer Interaction at QMW and Head of the Department of Computer Science. He holds BSc and PhD in Psychology. His past research has developed the theory and methods of Task Knowledge Structures and of user interface design in ADEPT. His present research is concerned with developing principles of Human Computer Interaction to support design and evaluation. He teaches undergraduate and masters courses in graphical user interface design, user interaction modelling and interactive system design.
Media transformations for the representation and communication of multimedia production activities

S. J. Morris  
City University,  
Department of Information Science,  
Northampton Square,  
London EC1V 0HB, U.K.  
Tel: +44 171 433 8344  
Fax: +44 171 433 8584  
E-mail: sjm@soi.city.ac.uk

Abstract
Case studies made of multimedia document production highlight the need for a means of classifying and describing the transformations of media elements which make up this process. The classification set out here contains twelve types belonging to two categories, constructive and supportive. A set of transformation representation rules provides the framework for succinct communication between production participants. This communication can serve both a descriptive function, describing events as the basis for design rationale, or a prescriptive function outlining detailed stages before they are activated.

Keywords  
media transformations, multimedia, production process, communication

1 INTRODUCTION

The work reported here constitutes the results from one section of a project whose primary objective was to provide a ‘production method’ for multimedia documents akin to the ‘design methods’ which are already part of software engineering. The general theme of this project was the integration, within one method, of the activities involved in the ‘design’ of document content and those required for the ‘development’ of software that provide structure and access mechanisms. The project reached positive conclusions regarding:

a) the usefulness of two new concepts, navigable discourse structure and media transformations,
b) the possibility of combining these concepts in a new discourse driven model of the production process, which provides a sound foundation for practical guidance, and,

c) the effectiveness of a staged method for the production of multimedia documents which uses this model as a framework.

An overview of the whole project appears elsewhere (Morris and Finkelstein, 1996). This paper concentrates on the concept of media transformations, a new means of describing and prescribing changes to media elements during multimedia document production. Media transformations offer a means of communication between production participants responsible for either activity.

The approach taken in the main project combined two complementary areas of study. One part of the work was more theoretical, the examination of concepts and theories relevant to the design and development of multimedia. The other part was more practical, the experimental production of multimedia documents using the standard multimedia tool, Director (Macromedia, 1994). The intention of this second part was to examine practical problems of production and, in the absence of any systematic studies, to elucidate the production process. It was from this second part that the notion of media transformations arose. The purpose of these experimental productions was to recreate live demonstrations of two software engineering tools, the Viewers (Nuseibeh and Finkelstein, 1992) and the System Architect’s Assistant (Kramer et al., 1993), as freestanding multimedia documents. The Viewers is a prototype environment supporting the framework known as Viewpoint Oriented Software Engineering (VOSE) (Nuseibeh & Finkelstein, 1992). The first case study produced the Viewersdemo. The second case study involved the production of the System Architect’s Assistant Demonstration (SAAD).

A live demonstration of a software engineering tool, such as the Viewers, and a computer based recreation of such a demonstration, for instance the Viewersdemo, illustrate the differences between multimedia and multiple media propounded in the project. Both are composite media objects, with different structures but the same communicative purpose, which is to show what the Viewers does and why it is important. The live demonstration is an example of multiple media. The demonstration incorporates several abstract media including still and moving images, graphics, natural language speech, computer languages, and text. The carriers for these abstract media include printing on paper, sound waves and the virtual medium provided by the computer. The combination of all these physical and abstract media takes place at the time of the demonstration and at the discretion of the presenter. The multiple media are held together by the actions and utterances of the presenter, who provides co-ordination by the selection and timing of these elements. The Viewersdemo is an example of multimedia. It uses the same abstract media as its live predecessor, but they are carried together in a physical medium
holding digital signals. This single physical medium is created, stored, manipulated and displayed exclusively in the computer. Integration of the abstract media takes place at the discretion of the developer and the resulting structure determines the manner of presentation to the audience.

Practical experience of producing these multimedia demonstrations highlighted the need for a way of classifying and representing activities. Such a representation would facilitate the planning of production, its subsequent recording and the continuing communication between participants. The crucial factors are the order and nature of the activities involved in the production process.

Knowledge about current and previous document states provides the basis for decisions about the order of operations. In the Viewer demo there was an initial listing of the potential sources for media elements and the creation of new versions of these sets of data was recorded, but without any precise indication of the activities that caused changes to come about. The lack of a suitable basis for a 'design rationale' (Carroll and Moran, 1991) was the cause of this deficiency. A general classification of possible multimedia production activities would offer a useful foundation. It would provide both a means of describing existing ad hoc methods and a means for defining new and more systematic approaches. The lack of such a means of describing activities seriously inhibited the logging of the production of the Viewer demo. In general it also forestalls any attempt to construct any general approach to systematic design.

A preliminary, static and three-level model of production provides an initial context for examining media transformations. Any method must allow development to take place on at least three different levels (Morris and Finkelstein, 1993). These are the levels of discourse structure, media composition and disposition, and presentation, shown diagramatically in Figure 1. Work may progress sequentially, concurrently or in some set of temporal combinations according to the method proposed. The arrows shown on Figure 1 indicate the possibility of simultaneous instantiation of related elements on different levels. The level of discourse structure is the most abstract. It is here that any underlying communicative purpose of the artifact acquires a coherent form. Choice, disposition and composition of individual media elements take place in the media composition and disposition level. The level of presentation involves both the design of the appearance of the final artefact and the finalisation of its structure as a computer based object.
2 DYNAMIC CHANGE VIA MEDIA TRANSFORMATIONS

Media transformations can introduce an essential notion of dynamic change into the static description of multimedia document production already given. Definition, at the level of media disposition and composition, of the media elements available will define the current state of this part of the document, whether interim or final. The questions left open are how movement between such states takes place and what the process of change involves. Similar questions apply at the presentation level of the static model.

In order to elucidate these problems, the earlier model is now extended to show two new features, the components generated at each level and the relationships of components between levels. A diagrammatic view appears as Figure 2. A discourse structure of a general type appears as an hierarchy with internal compositional relationships (shown as broken-line arrows). The abstract media selected and generated at the next level have an external relationship with the discourse components (shown as double-headed arrows), representing each singly or in some multiple combination. These media elements are then presented, individually or jointly, in a spread carried on the virtual physical medium, or may be attached to it as one of the access operations which are themselves also abstract media elements (relationships shown as arrows).

The process of change between intermediate document states involves a series of media transformations, each drawing on elements from earlier stages to create modified or new elements. Figure 3 illustrates diagramatically how two sets of transformations, \( T_1 \) and \( T_2 \), relate three states, 0, 1 and 2, of a document in production. The unconnected media element in state 1 indicates that design is in an ill-defined or intermediate state requiring attention.
Figure 2 Relationships between levels of earlier production model (Figure 1).

Figure 3 Transformations between document states.

Transformation processes are already clearly identifiable in the multiple media production techniques such as film production (Bloedow, 1991) and traditional cartoon animation production (White, 1986). As part of these multiple media techniques transformations take place between different physical as well as abstract media, for example the transfer of character drawings to film as well as the drawing of character sketches on the basis of a script outline. In the case of multimedia, preliminary stages may employ elements held in other physical media pending transfer to the digital.

Following such early transfers between physical media, it is assumed that the transformations that act on abstract media elements are executed by the production participants, with some assistance from tools that allow the manipulation of digitally based elements. Only a partial set of transformations will be within the technical capabilities of automatic digital tools presently available or envisaged.
Thus multimedia production can be seen as a series of transformations executed by the designers and developers participating in the process.

Revisions to the earlier production model allow the incorporation of specific media elements and their relationships with other components. The new version provides the context for media transformations, executed by the process participants, to function as the essential means for moving from artefact state to artefact state. In its final form the process model incorporates a fourth stage involving the construction of a navigable discourse structure as a means of defining the actual discourse structure of the document. As such this navigable discourse structure can be related to the intended discourse structure generated initially.

3 CATEGORIES OF TRANSFORMATIONS

Transformation activities require the manipulation of media elements. Transformation involves the generation or regeneration of one or more abstract media elements of the same or different types held on the digital medium, or possibly on some alternative medium during early cycles of design. In any state an artefact or document will incorporate a number of discourse components. The disposition of these components to one media type or another, the internal composition of the media elements, and their manner of presentation will be the result of some set of transformations performed upon elements in earlier states. The nature and number of these elements will determine the possible transformations that may be executed in order to reach the next state. The transformations fall into the two general categories of constructive and supportive transformations. The former facilitate the primary processes that directly result in elements of the final artefact, the latter facilitate subsidiary activities involving the use of elements in some kind of supporting or subsidiary role.

Manipulation of an initial text used for a multimedia document such as the View Demo offers simple examples of both categories. Transcription of the original demonstration commentary from a sound recording to a text involves a constructive transformation because at least some part of this text is likely to survive within a text element of the final presentation. At the same time this transcribed text forms the basis for a new spoken commentary which will also be part of the final document. Until this new commentary is recorded, the transcribed text will substitute for it within the production process whenever it provides essential content information relevant to the production of other media elements, such as the capture of images from the running application. Use of this transcription in such a subsidiary role represents a supportive transformation.

The supportive category also introduces a notion of purpose. It is open to production participants to decide the possible uses of a particular element, thus determining whether it is used when a particular supportive transformation takes
place. This opens the way for transformations to be used prescriptively as well as descriptively. In the last example, for instance, a participant fixed the text as the basis for the final spoken commentary by defining its generation specifically as the creation of a substitute for that commentary.

The necessity for such functionally supporting roles is one reason why transformations do not necessarily take place singly. New individual elements may be the result of more than one constructive transformation, for example the revision and merger of two graphic elements, or some combination of constructive and supportive. The notion of combining transformations is examined further below.

4 DEFINITION OF TRANSFORMATIONS

This section provides definitions of the specific transformations within each category, accompanied by general examples of their products {shown in brackets}. The analyses of animation production and the viewer demo which follow provide detailed examples of their use singly or in combinations as a means of production description.

Constructive transformations

*origination*: the initial creation of a single medium element, without specifiable input sources or the designation of a primary source {A statement of an initial concept or a primary source document},

*amplification*: expansion of one element to form one or more elements in the same medium {Addition of new features to a diagram},

*revision*: an element (or elements) in one medium supplanting an element in the same medium via any type of revision, alteration, or subdivision {Alteration of images after comparison with what is to be an accompanying text},

*translation*: an element in one medium replacing an element in another {Moving images described in text},

*outline*: abbreviated or précis version of an element based upon another existing or as yet unrealised element in the same medium {List of functions shown in a software demonstration},

*merger*: combination of two elements of the same medium to form a third, also of the same type {Any composite image created from more than one source},

*amalgamation*: combination of at least two elements of different media types in a composite element retaining individual identities but with combined purpose {Text and images within a multimedia spread},

*proxy creation*: creation of an element in one medium to stand in place of an element in another medium pending the creation of that second element and representing some essential characteristics of it {Text of spoken commentary},

*substitute creation*: creation of an element in one medium to stand in place of an element already realised in another medium, and representing some essential characteristics of the element {Any representation of timing information}.
Supportive transformations

proxy use: employment of proxy already created {Image generation guided by text of expected commentary},

substitute use: employment of substitute already created {Use of a timing diagram to support any constructive transformation},

comparate use: use of an element as the basis for a comparison between it and another element, with a view to some constructive transformation to the latter {Fixed image sequence used to check text}.

5 ANALYSIS VIA TRANSFORMATIONS

This section shows the descriptive function of media transformations, defined in the previous section, as applied both to traditional multiple media and to multimedia production.

An analysis of the production sequence for animation shows how these transformations can be applied to traditional multiple media development. In this example, shown in Table 1 below, the stages (0 - 6) are the first seven of the sixteen detailed by White (White, 1986); the remainder are Line tests, Clean up, Trace and paint, Backgrounds, Checking, Final shoot, Rushes, Dubbing and Answer print.

In different ways the application of these transformation categories to multimedia production is both simpler and more complicated than their application to multiple media. It is less complex because, except in the preliminary stages, it is less important to take account of any changes of physical media that may be taking place. Although omitted from the preceding analysis of animation production, consideration of these carrier media would be essential for any full description of the process from the point of view of design. The different materials and techniques required, for example to prepare storyboards (Stage 2 in Table 1) and final images on acetate (Stage 9), also determine the skills which the participant directing production, fulfilling any editorial role, must co-ordinate.

On the other hand multiple media production is simpler because it need not be concerned with access operations, these being predetermined by the mechanisms used for manufacture and display of moving images on photographic film. The multimedia document will have no such standard access operations, making their choice and implementation an essential part of production, and making the application of transformation categories more complex. The analysis of the Viewer demo production which follows shows how this can be done.
The general order of production activities for the viewerdemo followed five stages:

0 - 1 Establishment of content,
2 - 6 Creation of two sets of key media elements,
7 Construction and testing of access mechanism,
8 - 11 Merger of key media elements and creation of additional media elements,
12 - 14 Integration of media and mechanism and their testing.

---

**Table 1** Transformations within multiple media animation production (Stages 0-6)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Original concept</td>
</tr>
<tr>
<td>0.1</td>
<td><em>origination</em> of speech (or text)</td>
</tr>
<tr>
<td>0.2</td>
<td>outline created for final whole</td>
</tr>
<tr>
<td>0.3</td>
<td>initial <em>proxy created</em> for final whole</td>
</tr>
<tr>
<td>1</td>
<td>Script</td>
</tr>
<tr>
<td>1.1</td>
<td><em>translation</em> of speech in 0 to text</td>
</tr>
<tr>
<td>1.2</td>
<td><em>amplification</em> of text in 1.1</td>
</tr>
<tr>
<td>1.3</td>
<td><em>proxy</em> in 0 <em>used</em>, and displaced</td>
</tr>
<tr>
<td>1.4</td>
<td><em>proxy created</em> in 1.2 for speech, sound and moving images in final whole</td>
</tr>
<tr>
<td>2</td>
<td>Storyboard</td>
</tr>
<tr>
<td>2.1</td>
<td><em>translation</em> of speech in 0 into still images</td>
</tr>
<tr>
<td>2.2</td>
<td><em>translation</em> of text in 1 into still images</td>
</tr>
<tr>
<td>2.3</td>
<td><em>merger</em> based on 2.1 and 2.2</td>
</tr>
<tr>
<td>2.4</td>
<td><em>amplification</em> (and origination) based on 2.3</td>
</tr>
<tr>
<td>2.5</td>
<td><em>proxy created</em> in 2.4 for moving images in final whole</td>
</tr>
<tr>
<td>3</td>
<td>Soundtrack</td>
</tr>
<tr>
<td>3.1</td>
<td><em>translation</em> of text in 1 into speech and sound to form part of a principal component of the final whole</td>
</tr>
<tr>
<td>4</td>
<td>Track breakdown</td>
</tr>
<tr>
<td>4.1</td>
<td><em>translation</em> of speech and sound in 3 to graphics and text representation</td>
</tr>
<tr>
<td>4.2</td>
<td><em>substitute</em> created for soundtrack in 4.1</td>
</tr>
<tr>
<td>5</td>
<td>Character designs</td>
</tr>
<tr>
<td>5.1</td>
<td><em>translation</em> of text in 1 into still images</td>
</tr>
<tr>
<td>5.2</td>
<td><em>revision</em> and <em>amplification</em> of still images in 2</td>
</tr>
<tr>
<td>5.3</td>
<td><em>use of proxies</em> from 1 and 2</td>
</tr>
<tr>
<td>5.4</td>
<td><em>merger</em> based on 5.1 and 5.2</td>
</tr>
<tr>
<td>6</td>
<td>Leica reel</td>
</tr>
<tr>
<td>6.1</td>
<td><em>revision</em> and <em>amplification</em> of still images in 2</td>
</tr>
<tr>
<td>6.2</td>
<td><em>substitute</em> from 4 <em>used</em> in place of soundtrack</td>
</tr>
<tr>
<td>6.3</td>
<td><em>translation</em> of 6.1 into moving images using 6.2 as comparate</td>
</tr>
</tbody>
</table>
Table 2  Media transformations in the production of the \viewer\demo (Stages 0-5)

O  Original documents
0.1 origination (in some pre-production process) of published papers describing the \viewer\ [t,g,i],
0.2 origination (in some pre-production process) of transparencies used in seminar explaining VOSE [g,i,t].

1  Live demonstration
1.1 origination of recorded commentary [sp].

2  Transcription
2.1 translation from recorded commentary in 1 to text transcription [t],
2.2 substitute created for original commentary.

3  Screen dumps from repeated demonstration run
3.1 origination of set of screen dumps [i] from a run of the \viewer,  
3.2 substitute use of text transcription from 2 as a guide for the transformation in 3.1,
3.3 revision of text transcript [t,g] from 2 with embedded notes indicating position of principal screen changes.

4  Screen images for multimedia demo
4.1 revision of screen dumps [i] from 3,
4.2 revision of images etc. in transparencies [i,g] from 1
4.3 merger of individual dumps from 4 to form initial set of demo images [i,g],
4.4 substitute creation of demo image list [t]

5  Text keyed to images
5.1 revision of text from transcription in 2 to form script [t,g] providing marked blocks of text,
5.2 use of substitute image list created in 4,
5.3 amalgamation of text from 5.1 with image symbols from 4.4 to show relationship between images and text [t,g],
5.4 creation of proxy for final commentary [t]

In Table 2 letters in square brackets stand for the constituent media produced by each transformation [t ext, sp eech, i mage, g raphics, v ideo]. The computer applications involved, most importantly the \viewer itself and Director, might be included in this table, in terms of text or numeric files in non-natural languages, abbreviated to plt for programming language text; they are excluded because the purpose of the analysis is to show the transformations, their sources and products, not the means by which each came about. Table 2 details Stages 0-5; Stages 6-14
involved Demo mock-up, Access mechanism construction and testing, Formatted
script, Image and script merger, Commentary recording, Script with timing, First
integration, Test of first integration, Final integration.

Retrospective generation of this analysis, on the basis of a log which lacked any
systematic means of recording activities, shows the robustness of the categories
proposed and their usefulness in the context of multimedia production. This
exercise suggested that improvements might be made in two ways, by a more
succinct shorthand for recording and analysis and by a simple graphic
representation. The next section considers the first possibility; the second forms an
important part of on-going work.

6 RULES FOR TRANSFORMATION REPRESENTATION

Any representation for media transformations must be able to function as a means
both for recording production activity that has already taken place and for planning
the sequence of activity required to produce a particular document. The definitions
set in Section 5 above need extension in two ways in order to provide such a
representation:

~ identification of the sources and outputs required for each transformation,
~ specification of combinations of supportive and constructive transformations
  that produce new media elements.

The first of these extensions is essential for tracing the origin of media elements
in any record of production and the second for direction of any new production. The
set of transformation representation rules (TRR) shown in Table 3 uses Backus-
Naur Form to define a syntax for describing media transformations in a way that
meets these requirements. Reserved expressions appear in boldface type; constants
appear between single quotes; names and numbers to be fixed by the user appear in
italics; braces {} are used to denote groups which may be repeated zero or more
times; and a vertical bar | denotes choice.

Rules 1 - 4 derive from earlier definitions of media and transformation, plus two
elaborations: inclusion of program language text (plt) and other sign systems (oss)
to replace the ‘other’ media type, and a definition of ‘element compositions’ which
combine more than one media type. Rules 5 - 11 provide the means for identifying
the sources and outputs of transformations.

Rules 12 - 14 define higher level constructs. A complete transformation (14)
comprises either a primary transformation (12) or a primary transformation
supported by a subsidiary transformation (13). Primary and subsidiary
transformations involve specified media elements in constructive and supportive
transformations respectively. The subsidiary transformations may be composed of
multiple supportive transformations. The positioning of the symbol ‘>’ and the separator ‘,’ aids differentiation between primary and subsidiary.

---

Table 3 Transformation Representation Rules (TRR)

1. abstract_media_type ::= t | sp | s | g | i | v | plt | oos
2. element_composition ::= '[' abstract_media_type
   { ',' abstract_media_type } ']
3. constructive_transformation ::= orig | amp | rev
   | trans | outl | merg | amalg | cprox | csubs
4. supportive_transformation ::= uprox | usubs | comp
5.1 orig ::= 'orig >' primary source
5.2 amp ::= source_reference 'amp >' output
5.3 rev ::= source_reference 'rev >' output
5.4 trans ::= source_reference 'trans >' output
5.5 outl ::= source_reference 'outl >' output
5.6 merg ::= source_reference source_reference
   { source_reference } 'merg >' output
5.7 amalg ::= source_reference source_reference
   { source_reference } 'amalg >' output
5.8 cprox ::= proxy_name 'cprox >' source_reference
5.9 csubs ::= source_reference 'csubs >' subs
6.1 uprox ::= 'uprox' source_reference '>
6.2 usubs ::= 'usubs' subs '>
6.3 comp ::= source_reference 'comp' source_reference '>
7. output ::= output_ref_number media_element_name
   element_composition
8. subs ::= output_ref_number substitute_name element_composition
9. primary source ::= output_ref_number
   name_of_primary_source element_composition
10. output_ref_number ::= stage_number output_number
11. source_reference ::= output_ref_number
12. primary ::= constructive_transformation
   { ',' constructive_transformation }
13. subsidiary ::= supportive_transformation
   { ',' supportive_transformation }
14. complete_transformation ::= primary | subsidiary '',' primary

Use of TRR provides a more succinct description of the production process, defining more clearly the relationships between activities, their sources and
products. Production of the viewer demo can now be recorded fully in the form shown in Table 4. Output reference numbers are highlighted using boldface type to aid backwards traceability. This visual aid is useful because the syntax does not guarantee any specific mapping between sources and outputs of different transformations.

Table 4  Production of the viewer demo following TRR

| orig > 01 | published papers [t,g,i] |
| orig > 02 | seminar transparencies [g,i,t] |
| orig > 11 | recorded commentary [sp] |
| trans > 21 | transcription [t] |
| csubs > 21 | transcription [t] |
| usubs 21 > | orig > 31 | screen dumps [i] |
| 31 comp 21 >, 21 rev > 32 | transcript + screen change notes [t,g] |
| 31 rev > 41 | screen dumps [i] |
| 02 rev > 42 | transparency images [i,g] |
| 41 42 merg > 43 | initial set of demo images [i,g] |
| 43 csubs > 44 | demo image list [t] |
| usubs 44 >, 21 rev > 51 | script in marked blocks of text [t,g], |
| 51 44 amalg > 52 | script + image symbols [t,g] |
| final commentary cprox > 52 [t] |
| uprox 52 >, 52 comp 43 >, 43 rev > 61 | demo mock-up images [i,v] |
| 61 outl > 71 | dummy elements [i,v,t,sp] |
| final elements cprox > 71 |
| orig > 72 | access operations [plt] |
| uprox 71 >, 71 72 amalg > 73 | test demo [i,v,t,sp,plt] |
| complete demo document cprox > 73 |
| uprox 73 >, 72 rev > 74 | tested access operations [plt] |
| 52 rev > 81 | script in final presentational form [t] |
| 61 81 amalg > 91 | first text/image combination [t,i,v] |
| 91 rev > 92 | text inconsistency check [t,i,v] |
| 92 rev > 93 | image inconsistency check [t,i,v] |
| 93 trans > 101 | recorded commentary [sp] |
| 101 outl > 111 | timing information [t,g] |
| 111 93 amalg > 112 | script + timing information [t,g] |
| final commentary cprox > 112 |
| 74 93 101 amalg > 121 | first integration [i,v,t,sp,plt] |
| uprox 112 >, 112 comp 121 >, 121 rev > 131 | final revision [i,v,t,sp,plt] |
7 CONCLUSIONS

The novel concept of media transformations provides the means of showing how the document structure may move between static states as its scope and form are extended. The differentiation made between constructive and supportive transformations recognises that, among the interim products of production, some contribute directly to the final document, while others act in a subsidiary capacity. The media transformation that are defined provide a comprehensive guide to production activities and a detailed means of describing, communicating and proscribing the production of any particular document. Analyses of animation production and the document produced in the Viewer demo case study, show the applicability of the concept to both multiple media and multimedia. A set of transformation representation rules (TRR) in BNF enables the succinct description of production. Composite statements will show the complex combinations of source elements and transformations that may result in a single new media element. The major concern of future work will be the visualisation of production in terms of transformations, providing a graphical representation to ease communication between participants for all purposes.

8 ACKNOWLEDGEMENTS

The author acknowledges gratefully the overall guidance and assistance of Anthony Finkelstein and the financial assistance of the Engineering and Physical Sciences Research Council of the UK during the time of this research.

9 REFERENCES


10 BIOGRAPHY

Stephen Morris originally studied the history of art at Cambridge University and subsequently followed a career in town planning. He later returned to higher education and obtained his doctorate at the Department of Computing of Imperial College. He now holds the post of Lecturer at City University in London. His research interests include the investigation of the theoretical and practical problems associated with the cooperative production of multimedia documents, in particular the implications for software engineering of integrating the separate activities involved in content generation and software development. He also works on representational languages for object-oriented software development.
Part Two

Supporting Multimedia Design
Finding the “Right” Image: Visualizing Relationships among Persons, Images and Impressions

K. Nakakoji\textsuperscript{1,2,3}, Y. Yamamoto\textsuperscript{3}, K. Sugiyama\textsuperscript{3}, S. Takada\textsuperscript{3}

\textsuperscript{1}PRESTO, Japan Science and Technology Corporation (JST),
\textsuperscript{2}Software Engineering Laboratory, SRA Inc.,
\textsuperscript{3}Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma, Nara, 630-0101, Japan
tel: +81-743-72-5381; fax: +81-743-72-5319
e-mail: kumiyo@is.aist-nara.ac.jp

Abstract
In designing multimedia information, authors need to use images that best represent the design intention. The use of images, however, can be misleading. Images can be interpreted in many ways, and the same image may be thought of as pretty, cute, childish, or immature depending on the context. Due to this situation, searching for images from an image library becomes a challenging task because authors cannot specify what they are looking for without understanding what impression people have with each image. The goal of the research presented here is to support casual authors in finding images for their authoring task by understanding (1) how their task is best represented using affective words, and (2) what impressions people have with images. We propose a model to visualize individual differences in impression of images allowing users to explore the space of relationships. The model consists of a set of triplets of a person (P), an image (I), and an affective word that represents impression (W). We have built a computational environment EVIDII to visualize the relationship among P, I and W using two types of spaces. The paper concludes with a discussion of effectiveness of the approach.

Keywords
image search, impressions, Kansei information, visualization

1 INTRODUCTION
The goal of multimedia information design is that meanings behind the information are well-communicated - the meaning is shared among audience as the author has intended (Nakakoji 1996). As more and more casual users have an opportunity to author multimedia information, we have found more cases where authors are
misusing multimedia (Forsythe 1998). Casual users do not have domain knowledge about the use of color, for example, and know little about what effects or consequences colored representations have on humans (Murch 1984). This results in multimedia information that miscommunicates the author's intention only because the author used colors in a wrong manner.

An image (computer graphics and pictures) is a multimedia domain that requires domain knowledge to be appropriately used in an authoring task. Images can be more ambiguous than text. For example, one image may be interpreted as a cute picture or a silly picture depending on the context. Factors that determine the context include topics of the information, purposes of the information, and most importantly, who receives the information. Professional multimedia authors have domain knowledge to understand or predict what effects images have. Casual users do not have such knowledge, and therefore, may keep looking for an image which is inappropriate for their authoring task.

Our goal is to support casual multimedia authors to find images for their authoring task that will be interpreted in the same manner by the audience as intended by the authors. There are two challenges in pursuing this goal. First, images can be interpreted in many ways, and we need to help users understand what impressions other people have for images. When an author wants to design a ‘refreshing’ homepage, an image selected for the page needs to be interpreted as ‘refreshing’ or as a similar impression; otherwise, the use of the image is ‘wrong’ in the authoring task.

Second, the goal of an authoring task can be vague and ambiguous, and fluctuate (Nakakoji 1996). Authors are not able to completely articulate what the goal is at the beginning of the authoring task as authoring tasks are ill-defined design tasks (Simon 1981, Fischer 1991). An approach to simply retrieve images based on what authors have specified as the initial requirements will not work.

Existing research in image retrieval based on ‘feelings’ or ‘meanings’ that represent an authoring task aims at developing a best mapping between words and physical properties of images (Kurita 1992, Inoue 1996, Hasegawa 1997, Isomoto 1996). This approach involves three issues. First, it is impossible to map everybody’s impression in a generic manner. It may be possible to have multiple associations for different groups of people, but it will not solve the problem. We have found in one of our user studies that even the same person has different impressions at different times for the same image. Second, words themselves give different impressions to different people. Different people have different connotations for the same textual expression; for example, one might think that the term ‘gorgeous’ reminds him of a beautiful woman, whereas others might associate ‘gorgeous’ with a more negative meaning, such as being overly expensive. Third, the approach does not take into account the fact that authors’ requirements fluctuate. Although authors may look for ‘refreshing’ images as they think their task is to develop a ‘refreshing’ homepage, they may later find that their task is better represented with the term ‘natural’ than ‘refreshing.’ Retrieving images based on a specific requirement as a one-shot affair will not suffice to support casual authors in finding images for an authoring task.
Instead of trying to train a machine to find the right mapping between words and images so that users can retrieve images based on a task specification, our approach is to use the machine to visualize the differences of impressions of images, and to allow people to find associations by exploring the visualized information spaces. The power of multimedia allows us to use the human perception skill on externalized representations instead of simply presenting numbers calculated as correlation factors among impressions and images (Zhang 1997).

We have developed a model that deals with three elements: people, images, and words. The (P,I,W) triplets are then viewed from each element: the person-based, image-based, and word-based perspectives. Each perspective is then mapped to a two-dimensional or three-dimensional space, such as the HBS-space for the image-based perspective. The EVIDII (Environment for Visualizing Differences in Individual Impressions) system has been developed based on this model. The system provides four different types of views to explore the relationships among persons, images and words by looking at how the triplets are distributed. In exploring the relationships that are visually represented, authors have a better understanding of what words better represent the goal of their authoring task, and what images better serve their tasks by looking at what others think of the images.

In what follows, we first present the EVIDII model. Section 3 describes the EVIDII system, the rationale for the design of the system, and also presents a brief scenario of how users interact with EVIDII. We conclude the paper with a discussion of implications of our approach and future directions.

2 THE EVIDII MODEL

The EVIDII (Environment for Visualizing Differences in Individual Impressions) model uses the three elements of persons, images and words to denote and visualize relationships among them. We first give an overview of the model, then describe the three spaces that are used in the model, as well as the views which are used by the users.

2.1 Overview of EVIDII

The EVIDII model uses the three elements of persons, images and words to represent the space of association. S(P,I,W) represents a set of triplets \{(p, i, w)\} where p is a person identifier, i is an image identifier, and w is an affective word, such as 'clear', 'soft', or 'cute'. For example,

(Jack, Image#31, refreshing)

represents 'Jack thought Image#31 as refreshing'. Such data is collected using a questionnaire to ask which image is associated with which words.

The goal of the model is to allow people to explore how the three elements are related to each other. For example, authors should be able to ask questions such as:

- "In addition to Jack, who else found Image#31 to be refreshing?"
- "Are there any other images that Jack thinks refreshing?"
- "Does Jack find the image just 'refreshing'?"
Thus, EVIDII is an environment that visualizes relationships among the three elements so that the users of the system would be able to answer such questions by interacting with the system.

2.2 Basic Spaces
The EVIDII model provides three basic kinds of spaces using each of the three elements in the model as the base element: image-based space, word-based space, and person-based space. This approach of focusing on one element at a time was taken because none of people, images, or words can be linearly ordered in any understandable way.

When visualizing differences in relationships among persons, images and words, the actual representation of each basic space is also important, as it will affect the user’s interpretation (Zhang 1997). Each of the basic space can be visualized in the following way:

**Image-based space.** There is a number of ways to map images onto a two- or three-dimensional space. For example, a system can compute the most frequently used color in each image or compute the RGB or HBS color coordination values.

**Word-based space.** To determine the physical location of the impression words, one way is to perform a survey asking people how ‘close’ pairs of words are, and compute the relative distance between each pair of words.

**Person-based space.** A personality test can be used to map people onto a two- or three-dimensional space.

As described in the next section, the EVIDII system based on this model has taken the HBS representation as one type of visualization using image as the base element. The NCDR-Word space, which is designed by the Nippon Color and Design Research Institute, is chosen for visualizing the word-based space. The person-based space has not been implemented in the current EVIDII system. In what follows, we focus only on the former two basic spaces in accordance with the current implementation of EVIDII.

2.3 Views
Two views each are provided for the image-based and word-based basic spaces. Table 1 summarizes the spaces and views, while Figures 1 and 2 represent the relationships among the basic spaces and views, and how the relationships among the three elements, persons, images and words, are visualized in the EVIDII model.

The image-based space (Figure 1) offers the ‘image-based word view’ and the ‘image-based person view’. In the word view (Figure 1-(a)), one can focus on a certain word, and examine the persons who associated that word to each image. For example, we can examine how everyone associated the word ‘refreshing’ to
each image. In the person view (Figure 1-(b)), one can focus on a certain person, and examine the impressions that the person associated to each image. For example, we can examine how Jane associated the words to each image.

In the same way, the word-based space (Figure 2) offers the 'word-based image view' and the 'word-based person view'. In the image view (Figure 2-(a)), one can focus on a certain image, and examine the persons who associated that image to each word. For example, we can examine how everyone associated impressions to Image#10. In the person view (Figure 2-(b)), one can focus on a certain person, and examine the images which are associated to each word by that person. For example, we can examine how Jane associated images to each word.

Table 1 Summary of Spaces and Views in EVIDII

<table>
<thead>
<tr>
<th>Basic Space</th>
<th>View</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Based</td>
<td>word</td>
<td>who selects which images for a word w?</td>
</tr>
<tr>
<td>Word Based</td>
<td>person</td>
<td>what words are associated with each image by a person p?</td>
</tr>
<tr>
<td></td>
<td>image</td>
<td>who selects which words for an image i?</td>
</tr>
<tr>
<td></td>
<td>person</td>
<td>which images are associated with each word by a person p?</td>
</tr>
</tbody>
</table>

Figure 1 The Image-Based Space and Word- and User-Views. The figure represents the transitions by (1) focusing on the user Jane, and (2) focusing on the word ‘refreshing’.
Figure 2 The Word-Based Space and Image- and User-Views. The figure represents the transitions by (1) focusing on the user Jane, and (2) focusing on the image that Jane thought to be ‘refreshing’.

3 THE EVIDII SYSTEM
This section describes the EVIDII system which is based on the EVIDII model. In this section, EVIDII will refer to the EVIDII system rather than the EVIDII model.

3.1 Overview of the EVIDII System
The current implementation of EVIDII supports two basic spaces: image-based space and the word-based space. EVIDII consists of the following components:

- Data collection interface
- HBS space (image-based space)
- Word space (word-based space)

The data collection interface of EVIDII is used to input data that are used in the HBS space and Word space. Given a set of images in GIF format and a set of impression words, EVIDII asks a user to associate words with each image, or vice versa.

The HBS space (Figure 3) and NCDR-Word space (Figure 4) allows a user to explore the space of relationships among persons, images and words. Thumb wheels on the left, right, and bottom of the windows can be used to rotate and move the space to better see the parts of the space that the user wants to see.

Figure 3 presents the HBS space. In the figure, there are twenty images allocated to positions in the three-dimensional space according to the values of Hue, Brightness and Saturation of the color used most in each image (Figure 3-(a)). By selecting one of the affective words in the space, the HBS space displays who associated that word to each image (Figure 3-(b)). Users can go back and forth between the two views. By selecting one of the users, the person view of the HBS-space shows the impressions that the user associated to each image. For example, in Figure 3-(c), Jack’s impressions are displayed in the HBS space.
Figure 4 presents the NCDR-Word-space. 174 words are allocated in the two dimensional space according to the word-space defined by the Nippon Color and Design Research Institute (Figure 4-(a)). The two dimensional space is represented in the cool-warm (x-axis) and soft-hard (y-axis). By selecting an image, the system displays the image-view, which shows who associated which word to this particular image (Figure 4-(b)). Users can go back and forth between these two views as well. By selecting a person (person-view), the Word-space displays what images this person associated with each word (Figure 4-(c)).

Figure 3 HBS Space and its Person- and Word-Views of EVIDII.
Figure 4: Word Space and its Person- and Image-Views of EVIDII.

3.2 Example Usage Scenario
This subsection presents a typical usage scenario using EVIDII.

Suppose Jane, a Biology student at a university needs to author a home page for her research group retreat. This home page should encourage people to participate in this retreat. Since the retreat is held in a mountain, she first wants to use an image related to nature.

Her research group members had already filled in EVIDII questionnaires to associate images with impressions so that students can use EVIDII to find appropriate images for their home page authoring task. Jane now starts using EVIDII to explore images that are ‘natural’.

First she uses the word view of the HBS space to see what images are thought of as ‘natural’ by whom (Figure 3-(b)). The system shows that Jane and Jack, her senior researcher, selected the same image of a stream in a woods as ‘natural’. She becomes interested in how Jack thinks of other images. She selects the person view of the HBS space for Jack (Figure 3-(c)). Now, EVIDII displays the affective words that are associated with each image by Jack. In the display, she finds an image of a river with a green bank adjacent to the one for ‘natural’, with which Jack associated the word ‘refreshing.’
This reminds her that ‘refreshing’ might be a more appropriate term to represent her intent for her authoring task. She looks at the person view of the NCDR-Word space to see what other images that Jack thought of as ‘refreshing’ (Figure 4-(b)). Jack had selected a couple of other pictures as ‘refreshing’. She likes an image of a meadow with blue sky. She changes to the image view of the NCDR-Word space to see how other people think of this image (Figure 4-(c)). There, she finds that her advisor Bob associated the term ‘natural’ for this image. She becomes quite sure about the image and decides to use that image for her authoring task.

3.3 Discussion
The scenario presented above indicates quite interesting possibilities for the EVIDII system.

- One can know how other people think of an image one is interested in.
- One can explore what image is thought of in a certain way by how many people.
- One can know what tendency images have in terms of the physical characteristics of the image.
- One can find other persons who have a similar or opposite tendency from oneself.

EVIDII supports people to understand what people think of an image. The system provides visual representations through which people can compare differences of impressions rather than using symbolic computation to represent similarity indices using algorithms such as the fuzzy logic or statistics.

Not only allowing people to understand what other people think of images, EVIDII supports users in understanding what they really need. As illustrated with the scenario, an author may not have a clear understanding for a requirement, as Jane first thought that she was looking for a ‘natural’ picture, and later changed her search to a ‘refreshing’ picture. Such situations are not supported in existing Kansei information systems and other types of image retrieval. Systems may be able to retrieve ‘natural’ images for Jane, but cannot support the refinement of her requirements. By allowing users to visually explore relationships among persons, images and words, EVIDII gives users an opportunity to see ‘what other people think of this image.’

Integration of multiple views with multiple element basis and the visual spaces allow people to explore the relationships among persons, images and words through related elements.

4 FUTURE WORK
This paper presented the EVIDII model and the EVIDII system as our approach to support casual users in selecting images in their authoring tasks by visualizing relationships among persons, images and impressions.

Ongoing and future work include:
Distributed EVIDII. We are currently working on implementing EVIDII on Internet using VisualWave. This will enable us to collect through the network data from a large number of people from a variety of cultures, and will allow such people to use EVIDII for their authoring tasks. We may be able to identify cultural characteristics of people-image-impression relationships. We may even be able to visualize culture (not necessarily geographic culture, but age groups, professions, religions, etc.); people from a certain culture think blue images as noble, for example.

Survey-setup Interfaces. When EVIDII is distributed and accessible over the network, users should be able to set up a set of images and affective words to make a survey to produce the person-image-word set. This might be more appropriate for a small group of people working together, for example, to collaboratively design a homepage. One can suggest tens of images and set up an environment for a discussion to develop a shared understanding about the image selection process.

Changeable Scales. Current implementation of EVIDII uses the HBS space and the NCDR-word space suggested by the Nippon Color and Design Research Institute. This, however, does not mean that these spaces are the best ones to visualize the relationships among persons, images and words. Rather, they should be thought of as an instance of mapping schemes. Users should be able to select a space of their choice, for example, users can choose for the image-based space the HBS of the most frequently used color, or that of the RGB, etc. This will allow people to explore which space best represents a certain tendency.

5 CONCLUSION
EVIDII helps people understand what other people might think of a certain image through exploring image-based and word-based spaces. This addresses the issue that people associate different meanings to the same multimedia representations. EVIDII also helps people refine their understanding of the task itself. By exploring words and images associated with each other through other people’s eyes, the system supports authors by having them become aware of new words that better describe their task. This word, then can be used to search for images that better match their authoring task.

We depend on the power of multimedia representations by visualizing the relationships. Much of what EVIDII offers do not necessarily have to be visualized. For example, if we wanted to find what other people think about a certain image, a simple list would suffice. Our approach here, however, is to make the best use of multimedia representations by using human perception skill (Norman 1986, Zhang 1997). The power of external representations have been underutilized (Yamamoto 1998). The work presented here is an attempt to use the power of visualization, which is not offered by simply using numbers to represent interdependence between elements. When possible, representation that enhance the understanding should be used. In EVIDII, for example, the image-based space
enhances which images are similar in terms of physical attributes, while the word-based space enhances which images are similar in terms of impressions.

More and more people have an opportunity to author multimedia information, which can be seen as a metaphor of the 'computation era' of the past and the 'representation era' of now and the future. Computers are used not just for scientific computation, but also for showing many types of ideas and thoughts. By being faced with the shift from the 'computation era' to the 'representation era', we have to seriously consider what multimedia representations best represent the task at hand. This paper presents our approach to address the challenge not on how to design effective multimedia information, but on how to understand what multimedia representation effectively reflects the intention.

6 REFERENCES


7 BIOGRAPHY
Kumiyo Nakakoji, a research fellow at PRESTO Japan, is an Adjunct Associate Professor for the Cognitive Science Laboratory at Nara Institute of Science and Technology. Her research interests include considerations of culture, communication and creativity in systems and design. She has worked for Software Research Associates, Inc. (Tokyo, Japan) for the last twelve years as a senior researcher at the Software Engineering Laboratory. She received her BS from Osaka University (1986), and MS (1990) and PhD (1993) in Computer Science from University of Colorado at Boulder.

Yasuhiro Yamamoto is a PhD student at Nara Institute of Science and Technology. His research interests include computer support for writing as design. He received his BS from Kyoto University (1996) and MS from Nara Institute of Science and Technology (1998).

Kimihiko Sugiyama currently works for Mitsubishi Electric Corporation. His research interests include Kansei information systems. He received his BS from Seikei University (1996) and MS from Nara Institute of Science and Technology (1998).

Shingo Takada is a Research Associate for the Cognitive Science Laboratory at Nara Institute of Science and Technology. His research interests include information search and object-oriented technology. He received his BS (1990), MS (1992), and PhD (1995) in Computer Science from Keio University.
Structuring multimedia data to facilitate decision making and reflection in product design

S. Phillips and J. T. McDonnell
University College London
Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK
Telephone: +44 (0) 171 267 1202, Fax: +44 (0) 171 284 2963
Email: S.Phillips@cs.ucl.ac.uk, J.McDonnell@cs.ucl.ac.uk

Abstract
This paper describes the construction of a multimedia design resource, structured to reflect the design process. Initially an issue based representation was used to organise the multimedia data for a complete season’s range of clothing in a sportswear company. This resource was used by the design team to review progress with the current season’s design and to reflect on the process overall once it was completed. Based on experiences with making use of this resource, we constructed a system for capturing and re-using design knowledge which better suits product designers’ priorities. We highlight the particular value of effectively structured multimedia material to design based companies where data is primarily visual in nature, where design cycles are rapid, and where brand consistency is important.

Keywords
Multimedia design resource, design rationale, decision support

1 INTRODUCTION
It is a widely held view that an ability to capture and retain knowledge used during product design is valuable to design based companies. The potential benefits of suitably structured multimedia resources could be enormous in design based industries where designers use old designs, samples and many diverse visual sources of material to inform and inspire their design activities (Phillips,
Much of the material referred to and created during product development forms an inherent part of the design rationale. This information may take many forms. As well as including data about previous products, competitors’ products, sales information and marketing directions, there may be video sequences which encapsulate the essence of the product in some way and story-boards containing pictures and media samples intended to illustrate the product direction and design proposals.

One advantage of structuring and retaining information on design decisions and influences is that changes in membership of the design team might be accommodated with less disruption. This is particularly important when designing branded goods, as it can be extremely damaging to be inconsistent with brand image. In design orientated companies it is common for many prototypes to be created during the product development cycle, many of these are discarded and the investment in their development is also lost.

At present, the knowledge and information used during each product development cycle is not retained for future use. One reason making it difficult to preserve such data is that it is difficult to structure it in a way that will remain accessible over time. Much of the knowledge used in design is also tacit and therefore by its nature difficult to express. Furthermore, product designers are not traditionally required to be explicit about the rationale behind their designs and therefore a substantial amount of the information that they use during the design process is not shared with others or formally recorded.

In this paper we describe the construction of a multimedia design resource organised around the issues addressed during the design of a product range.

2 DESIGN RATIONALE METHODS

Moran and Carroll (1991) identified a number of reasons for capturing and representing design rationale, namely to enable designers to reflect on their reasoning at different stages in the design process, to facilitate communication of the design rationale to others, to enable the design knowledge to be reused or reflected upon in future design projects.

A number of notations have been developed for representing design rationale. These notations have been created mostly for the domains of engineering and software design. Although there are differences between the design processes in these domains and that of product design, there are enough similarities for these notations to provide a suitable starting point for product design rationale capture.

Much of the current research on capturing design rationale is based on variations of the more broadly applicable Issue Based Information System (IBIS), first described by Kunz and Rittel (1970). An IBIS is intended to ‘guide the identification, structuring and settling of issues raised by problem solving groups.’ (op. cit., p.1). Its application to the context of design deliberation, takes the form of a network representation of the various alternatives that are
considered in response to issues and the arguments for and against the positions (Moran and Carroll, 1991).

MacLean et al (1996) developed a notation for representing design rationale called Questions, Options and Criteria (QOC). QOC is a notation for capturing what is considered during design which is also intended to encourage the exploration of design alternatives. A QOC representation is produced during the design process. It is based on the idea that a particular design is one which exists in a space of design possibilities and an analysis of the design space explains why the design emerged in the form it has from among the alternative possibilities. MacLean et al argue that creating a design space rationale ‘will repay the investment in its creation by supporting both the original process of design and subsequent work on redesign and reuse by providing an explicit representation to aid reasoning about the design ..... serving as a vehicle for communication, for example among members of the design team’ (op. cit., p.55).

It was not the intention in QOC to create a notation that would record the design deliberations per se, but rather to construct a representation of the design space that surrounds an object. The authors emphasise the fact that a QOC representation must be ‘carefully crafted’ itself and created along with the design specification. This makes it a potentially intrusive method on the designers’ time. The merit of conducting a design space analysis before commencing a design is that it provides a designer with a detailed analysis of the design alternatives. However certainly in product design, the time factor would often be prohibitive to such analysis. The IBIS notation not only requires less effort to construct but also does not require the designer to deviate far from the usual design tasks and provides a more natural representation of the design rationale.

Lee and Lai (1996) have devised a notation for representing decision rationale called DRL (Decision Representation Language). However they highlight the fact that there are limitations in the scope of DRL in so far as it is primarily a model for capturing decision making and it does not capture certain aspects of design rationale. This consideration and the complexity of the representation was also an inhibiting factor for us as we wanted to give the designers something relatively simple to work with initially.

QOC shares the same drawback of complexity with DRL, although less so. An IBIS like approach presents a fairly natural way of structuring a design rationale since it allows issues and their arguments to be presented as they are discussed and does not artificially force the designers to be explicit about criteria to which they would not otherwise have paid any attention.

In summary, three of the prominent existing methods for representing design rationale were evaluated: DRL, QOC and IBIS. We concluded that IBIS provided the most appropriate starting point due to the simplicity of its representation and the ability of the model to represent the natural process of product design. A further important advantage of the IBIS model is that it ‘is geared to capturing deliberation as it happens,’(Moran and Carroll, 1991, p.198) and it does not require the rationale to be constructed as a separate part of the development process.
3 CONSTRUCTING THE INITIAL KNOWLEDGE BASE

We conducted a detailed study to capture the design rationale for a Spring/Summer 1998 product range for an American Sportswear company. Subsequently an issue based structure was constructed using the data collected. The aim of this knowledge base was to store all the information relating to the products being designed and to capture the decisions relating to their development. An IBIS-like representation was used as an initial basis from which to investigate the real problems of design rationale in the field of product design.

The notation consisted of nodes and links to show relationships between them. The nodes represented issues, positions (ideas) about how to address the issues and arguments for and against the positions. These could be simply linked to one another in a logical manner. In addition, the notation allowed issues to be directly linked to other issues and positions to be linked to other positions. (This latter extension to IBIS was originally proposed by Conklin and Begeman (1988)). A decision node type was also incorporated into the notation which could be linked to issues or positions, this too is an extension of the original IBIS notation. In the remainder of this paper we refer to this notation as IBIS\* for simplicity.

To explore the requirements for useful representation (maps) of the collected design knowledge we used a software environment that supported a graphical representation of the design rationale and which allowed us to link multimedia data to the nodes of the maps. Each node could contain some explanatory text and could be connected to other reference files launched directly from the nodes, such as other IBIS\* maps, graphic, video or audio files. The resulting structured collection of data was installed on the computers belonging to the design team incrementally as it was being constructed. An analysis was then conducted to assess its effectiveness and limitations.

Showing the designers an insubstantial prototypical representation, comprising a small quantity of data to which they find it difficult to relate and asking them both to imagine a realistic, substantial set of useful data and to conjecture how they might make use of it, is asking too much. In order to gain quality reflection and responses from the designers the maps needed above all to be realistically complex, capturing as much knowledge as possible, thereby making them relevant in the real design setting. It was therefore essential that the representation should be comprehensive and include a full season of design data in order to make our findings credible. To achieve the necessary complexity and detail we followed a complete product development season. All of the formal design meetings were attended and fully transcribed, and further visits were made to members of the product development team to collect other background material that had not been gleaned in the formal design meetings. In addition detailed interviews were carried out to obtain feedback on the value of the IBIS\* knowledge structure and to assess its effectiveness.
The IBIS$^+$ knowledge structure contained approximately 200 image files, 4 video files, 2 audio files, and 30,000 words of design meeting transcripts. It consisted of 13 IBIS$^+$ maps containing 480 nodes and 471 links.

4 ANALYSIS OF THE INITIAL KNOWLEDGE STRUCTURE

The designers used the design rationale as it was being constructed during the design season, to reflect on what they had stated they would achieve at the initial meetings and to identify what decisions had for various reasons fallen by the wayside. At the end of the design season when the IBIS$^+$ maps were completed the product design team experimented with them for a few weeks. Then a detailed analysis of their reflections and opinions was conducted.

Concern was voiced regarding the ability to capture all the design decisions, despite the fact that as far as possible all the formal design meetings had been attended. Although things not explicitly expressed cannot be captured, one might expect that decisions of import would be shared or discussed in the formal design meetings. Unfortunately however, this is not always the case. For example design issues may be discussed between a sub group of the design team on the way to meetings or in other non scheduled discussions. Furthermore, urgency sometimes dictates that decisions have to be made outside of formal meetings. This may sometimes mean that there is no time to consult and inform the rest of the design team about some of the decisions made. This is an inevitable limitation of any attempt to capture real decision making. On a related issue Fischer has pointed out that, 'a truly complete account of the reasoning relevant to design decisions is neither possible nor desirable. It is not possible because some design decisions and the associated reasoning are made implicitly by construction and are not available to conscious thinking. Some of the rationale must be reconstructed after design decisions have been made. Many design issues are trivial; their resolution is obvious to the competent designer, or the design issue is not very relevant to the overall quality of the designed artifact. Accounting for all reasoning is not desirable because it would divert too many resources from designing itself.' (Fischer et al, 1996, p.270)

To address these inherent limitations we propose a reflective stage at the end of the design process where the decision structure is reviewed, corrected, annotated and updated in order for the knowledge representation to be of value for future reference. In our study we found that this task does not need to take more than a couple of hours and ensures the design rationale structure is up to date. It is also a useful exercise to refresh the memories of the design team about the process that has been gone through and the goals that have or have not been achieved.

One of the major comments was that it would be extremely useful for the current design rationale to be used as a template to start building the next season's range. However the maps editor was not at an advanced enough stage for the designers to be able to develop a template for themselves. There were many other important findings from the analysis of the designers' experience with using
our IBIS+ maps. These concerned weaknesses in the IBIS+ notation; how to support reflection during and after the design process; interface requirements – ways of presenting data from the maps for specific purposes: and direct benefits to the design team and the company of retaining knowledge used during the design process over a succession of design cycles.

The limitations of the IBIS+ notation led to a complete restructuring of the material collected for the whole design season. The adaptation of the notation and its extension to cope with the realities of the product design process are described in a separate paper (Phillips and McDonnell, 1998). A section of one of the maps which uses the improved notation, which we refer to as DR maps, is shown in Figure 1. In the remainder of this paper we discuss those issues concerning product design which most affect the way the multimedia design data is structured to be most useful.

![Figure 1](image)

Figure 1  A section of a design rationale map.

5 STRUCTURING MULTIMEDIA DATA INTO A TOOL TO SUPPORT DESIGN DECISIONS

Several issues were raised concerning the capability of the tool to support reflection on the design process, which in turn would have an impact on the design decision making in the next season’s product development meetings. It
quickly became apparent that there was a requirement for the knowledge structure to be used not only as a record of the design rationale but also to enhance strategic planning and to provide a starting point for the following season’s design cycle.

A particularly useful aspect was thought to be the use of the knowledge structure as a reflective tool to enable the design team to review the previous range development process and to highlight where goals had been achieved. The DR maps sparked debate within the design team as to why certain issues had remained unresolved or had fallen by the wayside. The design team were given access to the knowledge structure as it was being built, and while the design process was still ongoing. They noted that it was useful to use the maps to see what goals had been achieved and identify what had been accomplished and what had not, even before the design process had been completed. Figure 2 shows how visuals of products that are discarded from the range at an early stage are retained within the multimedia maps in order that the designers may refer back to them at later date. The knowledge structure provided an insight into the reasoning behind certain decisions being made and allowed the identification of the issues that were lost during the design process. The design team also noted that it would be a useful historical record, as over time catalogues and samples get mislaid.

Many claims are made about the importance of recording the design process, but in the field of product design little research has been conducted to substantiate this.

Figure 2  Visuals of discarded designs contained within the maps.
It was further commented that the knowledge structure provided a useful starting point for the next season’s range planning. It enabled a review to be quickly conducted of the goals identified at the beginning of the previous season and a new set of goals to be drawn up for the coming season. It became apparent that the decision structure would not change entirely from season to season. Some of the design decision structure would be unique each season, but a relationship would exist with the structure from the previous season. The design team wanted to use the current season’s DR maps as a template for the next season’s maps to provide a starting point and to reduce the work of constructing the next season’s design rationale. A set of DR maps would be used initially to identify a model path for the product development. From this a template could be created to form a basis for the following season’s maps. This would make the knowledge base a very useful tool for design/product managers and it would speed up and enhance the following season’s range planning. However, a reflective stage would be required first, in order to ‘tidy up’ and annotate the maps with decisions not made explicit in the design meetings. Given the correct tools this is a modest task, although it would need to be done as soon as possible at the end of the design process. The design team felt it would provide invaluable support for their strategic planning and decision making at the start of the following seasons design cycle. Figure 3 demonstrates how a template will be created after a reflective stage at the completion of a development season.

Another issue that was raised was that every time a design cycle has been completed, a link to the next season’s map should be added so that the design rationale can be traced through seasons. One reason that not all the rationale was captured in the DR maps was that many of the design scenarios understood by the design team, stemmed from previous season’s development. Over a period of time these omissions could be overcome to some extent by linking in seasonal progression. Exploration of the decision structure through a number of seasons would give an understanding of previous design scenarios and decisions.

Specific interface facilities were identified which would enhance the usefulness of the DR maps for making decisions about future seasons’ product development. Some of these were:

- Ability to select all products from the current range that are being carried over into the following season’s range (to import them directly into the new DR maps).
- Provision of a gallery summary of a previous range alongside a working screen for the current range.
- Slide show display of the path of evolution of any product selected from a gallery display.
Figure 3  Reflective stage adjustments followed by template creation.
As a source of reference data allowing historical exploration and as a decision making tool for the design team in years to come, the maps are currently presented with the final products as the end result in the decision trees that the maps represent. For use as a historical resource it is more sensible to be initially presented with the final range of products, and then to be able to go through the decision making process in reverse to see the evolution from the original inspiration. Graphics from each range should be viewed together or in categories of the various stage of product development, e.g. photos, coloured sketches, and line drawings. The designers prefer a slide show style display of the lines of development starting from the ends of the branches on the maps, so that any particular product can be taken individually and all the images along its development path can be viewed starting with the final photo of the finished product, as demonstrated in Figure 4.

![Diagram](image-url)

**Figure 4** Displaying the lines of development for each product.

In product design much implicit knowledge is visual data of a multimedia nature, which we believe can be usefully retained for later reference by setting it in context in a DR map structure. Figures 5 and 6 demonstrate the potential for capturing tacit knowledge by including visual data within the DR maps.
Figure 5  A clip from one of the inspiration videos contained with the maps.

Figure 6  A visual design story board within one of the maps.
6 CONCLUSION

The initial reaction of the design team to the DR maps was how well they graphically demonstrated the work conducted and the processes and decisions undertaken during the product development cycle. Having experimented with the resource, it became clear to the product development team that once the design rationale had been captured over a few seasons the benefits would become more significant.

Our research identified many positive benefits from constructing as full a representation as possible of the design rationale. These included:

- Aiding and speeding up the planning for the following seasons ranges.
- Ensuring higher quality decisions are made on the new ranges on the basis of reflection on the previous design season.
- Providing new team members with a history of previous product development and design decisions.
- Helping branded products to retain consistency in their brand identity.
- Allowing team members to analyse their own personal development.
- Providing an archive of previous design that will become increasingly useful with the passing of time as a resource for visual research/inspiration.
- Providing a resource for marketing activities that may incorporate 'retro' designs. Many branded clothing companies such as Levis use old, archived design work from previous decades in their 'lets return to our roots' marketing activities.
- Aiding quality reflection on both aesthetic and technical decisions. It is useful to explicitly represent design rationale in those product design domains where the over-riding decisions may be split into those which are aesthetic in nature and those addressing production/technical concerns. For example, a certain style may be adopted due to the design team's belief in it having the correct aesthetic values for the market and yet later prove to be a 'dog' in terms of sales. In such a situation if a design rationale representation had been constructed it might lead the design team to question their judgement on whether they were using the correct aesthetic values for the market place with this product. Similarly it would also be useful to have a rationale of technical/production decisions. In a case where a fabric proves to be unsatisfactory, resulting in a high number of product returns, the specification and rationale for choosing that fabric could be referred to when seeking a sensible resolution.

Ideally the design rationale would not be built by one person and then presented at the beginning of each meeting as it was in our work. We envisage it being constructed as an integral part of the design meetings by all members of the design team. Members of the team should be allowed to view or add to the rationale at any time and it should be used as a reference point during meetings.
for the team to reappraise goals and objectives. The nature of product design means that much of the tacit knowledge understood by the designers would be contained in the predominantly visual multimedia data linked to the design rationale structure. Although this data may not mean anything to the untrained eye, to an experienced designer it conveys a lot of valuable information and adds an expressive new dimension to formal design rationale notations. In clothing product design in particular, a range of similar products are created season after season, and there are two or three seasons each year. The amount of design re-use prevalent means that it appears to be much more practical to justify the effort of capturing design rationale in this sort of domain than in to the more traditional domains of software or engineering design.

More research is required to assess the feasibility of design orientated companies in general allocating adequate resources for capturing and retaining product development rationale over a substantial period of time. Nevertheless, it is clear from our research that if a company invests the time and money into collating and structuring design rationale they may be rewarded with a multimedia resource that will enhance their design decision making process. It would provide a more efficient approach to their product design and when collated over a number of design seasons could be used further to support their overall strategic planning.

7 REFERENCES


8 BIOGRAPHY

8.1 Sophie Phillips
Sophie Phillips is working on a PhD in the Department of Computer Science at University College London. She has a BA in product development from the London College of Fashion and has worked in the buying sector of the clothing industry. Her research interests focus on the development of information systems to provide design support during the product design and development process.

8.2 Dr. Janet McDonnell
Janet McDonnell is a senior lecturer in the Department of Computer Science at University College London. Her research interests include design support systems, knowledge management and strategic uses of I.T. for information management.
Creating the multimedia project experience database

J.W. van Aalst
Origin IT Services, Department Business Communication
P.O.Box 3336, Utrecht NL-3502-GH, The Netherlands, tel. +31 30 2911200, fax. +31 2911211
e-mail Jan-Willem.vanAalst@nl.origin-it.com

C.A.P.G. van der Mast
Delft University of Technology, Technical Mathematics and Informatics, Department information Systems
P.O.Box 356, Delft NL-2600-AJ, The Netherlands, +31 15 2782549, fax. +31 15 2786632, e-mail C.A.P.G.VanderMast@is.twi.tudelft.nl

Abstract
In this paper we present experiences and results of an inductive case study to identify the most fundamental project problems that are specific to the field of multimedia, with the aim of designing a framework for an online accessible multimedia project experience database. To identify the problems, we conducted 32 interviews on about 25 multimedia experts, gathering their experiences and opinions about success factors, knowledge numbers, management, communication, meetings, infrastructure, tools, etcetera. The results of these sessions are used in the design of a multimedia experience database from which multimedia experts can learn.

Keywords
multimedia projects, experience databases, improving project control
1. INTRODUCTION

The multimedia industry can be seen as a subset of the general IT-industry, the main difference being the incorporation of various new disciplines in multimedia project teams, and the use of new, innovative (and often unstable) technologies and development tools (Van Aalst & Van der Mast, 1996). These two aspects often cause considerable difficulty in producing high-quality multimedia systems such as computer based trainings, electronic performance support systems, marketing and sales presentations and internet sites (often sales kiosks) (see for example Van der Mast (1995)).

During the last few years, we see a slow but steady maturing process in the IT industry with the advent of the RAD methodology and its subsidiaries. Such methods, combined with the increasing power of software and hardware, offers companies better ways of climbing up the Capability Maturity Model scale (Humphrey, 1989). However, this maturing process cannot yet be found in a widespread fashion in the multimedia industry (England & Finney, 1996). Other than for example the film industry, where script writers comfortably work together with directors and video technicians (see for example Monaco, 1981), a graphical designer still does not communicate easily with a Java programmer, for example. Now that multimedia products are (rightfully) no longer the result of a team of programmers alone, there is need for a specific multimedia language or jargon, just as this has happened in the film industry. It will take several years for such a language to emerge (Laurel, 1993).

In the general IT-industry, various factors have been identified that cause problems during development projects (see Van Aalst & Van der Mast, 1995):
1. Sociological problems, consisting of organisational issues and communicational issues;
2. Technological problems, meaning the availability of robust tools, knowledge about, and experience with these tools, and a good working environment.
3. Political issues, where higher-level stakes influence the development process of the product.

We have summarised these problems in the following model:
Figure 1 Project problems in IT-projects

These problems are found in the multimedia industry also. Still, it seems that there are also problems that are very specific to the multimedia industry. For example, the various disciplines do not communicate easily with each other, and the HCI expert needs to communicate with virtually all disciplines. In this paper we identify these aspects and describe their role in the setup of an online accessible experience database to learn from past experiences with multimedia projects.

This paper is set up as follows. First, we describe the framework that we use to look at the area of multimedia project problems; then we describe the methods that we use to identify these problems; then, the actual experiment of identifying these problems and the setting up of the experience database is described. The paper then offers a brief discussion of the results, conclusions, and an overview of future research, as well as links to where more information about this research project can be found.

2. FRAMEWORK

An important way of improving an IT-product is to improve the way in which the project that leads to that product is organised; in other words, making sure there is sufficient control over that project at all times (improving control over the process). For the IT-industry, sophisticated 'planning & control' tools are emerging, like QSM's SLIM (Greene (1996), and Putnam & Myers (1996)) and Transform by SHL Systemhouse (Hughes, 1997). However, when examining such tools, we see that they do not yet support the specific nature of multimedia projects: they are still working too much from a computer-science point of view. For example, the tools described above can measure product size only in ELOC (effective lines of code), or in simple GUI units. In the multimedia industry, direct manipulation development tools are common; moreover, graphics, video and audio can be a substantial part of the total effort.

To adapt these tools to specific multimedia problems, one needs to have an overview of the experiences of multimedia experts, and their (interconnected) project problems, needs and wishes. To build that overview, we create, through interviewing sessions, a multimedia experience repository; a database that all multimedia experts can access
and can learn from. Such a database contains project experiences of project managers, administrators, visual designers, specificators, programmers, etc., about multimedia project problems such as project management, communication, meetings, customer participation, technical infrastructure, working conditions, project pressure, etc., as well as multimedia-specific knowledge numbers. In other words, there is both a quantitative (knowledge numbers) and a qualitative (emotional experiences) side to it.

We define the following roles, partly after (England & Finney, 1996):

- project manager
- project administrator
- quality assurance manager
- art director
- graphic designer
- interaction designer
- video artist
- video engineer
- audio artist
- audio engineer
- domain expert
- designer
- specificator
- programmer 3gl
- programmer 4gl
- tester

Multimedia experts can learn from each other’s experiences using the experience database. For example, when setting up a project bid, a multimedia project manager can check which parts of such a bid have in previous years often been the cause of questions or trouble. Or, a project manager can check what percentage of the budget his colleagues have reserved for making a project bid, for previous projects. Or, a programmer can check what technical pitfalls he or she should be aware of when making a Java applet. Moreover, as the use of the database increases, it can also grow to a level to where it is a mirror of all project problems in multimedia projects.

To make the multimedia experience database work, there would need to be a cycle of multimedia projects making use of the database, while at the same time storing new data in de database. This cycle is described in figure 2 below.

**Figure 2** The global framework of improving control over multimedia-specific problems
In this paper, we describe the use of the questionnaires to identify the multimedia specific project problems, their use in the design and realisation of the multimedia experience database, and the opinion of multimedia project members about the contents and usefulness of the database.

3. METHOD AND EXPERIMENT

The first phase in the process of setting up the cycle described in figure 2 has been realised. It consisted of an inductive case study of sixteen interviewing sessions with various multimedia experts. An initial questionnaire was designed, with an estimation of problems that would probably arise; on top of that, the experts were stimulated to explain their own opinions about the most fundamental problems. After that, a second iteration of fourteen interviewing sessions was held. In tables 1.1 - 1.3, we give a quantitative description of the interviewing sessions of the first iteration.

Table 1.1 Interview data for seven products

<table>
<thead>
<tr>
<th>Project type:</th>
<th>Duration (weeks)</th>
<th>Average # staff</th>
<th># interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBTraining</td>
<td>10</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CBTraining</td>
<td>12</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>CBTraining</td>
<td>17</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>presentation</td>
<td>24</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>presentation</td>
<td>8</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>publishing</td>
<td>20</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>EPSS</td>
<td>23</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

These are all multimedia products. CBT means Computer Based Training; EPSS means Electronic Performance Support System. Publishing means a kiosk application or other similar products.
Table 1.2 Interviews held for each of the project member roles

<table>
<thead>
<tr>
<th>Role</th>
<th># int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project managers</td>
<td>6</td>
</tr>
<tr>
<td>Designers/content experts</td>
<td>4</td>
</tr>
<tr>
<td>Graphic artists</td>
<td>2</td>
</tr>
<tr>
<td>Programmers</td>
<td>3</td>
</tr>
<tr>
<td>Testers</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1.3 Project phases in which the interviews were held

<table>
<thead>
<tr>
<th>Phase</th>
<th># int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>planning phase</td>
<td>6</td>
</tr>
<tr>
<td>design &amp; main build phase:</td>
<td>6</td>
</tr>
<tr>
<td>evaluation phase</td>
<td>4</td>
</tr>
</tbody>
</table>

The answers we got from these interviewing sessions were used for the design of an improved version of the questionnaire. These questionnaires were more specifically aimed at capturing the multimedia project experience that could be useful for future multimedia projects. For each of the three main phases: project bid/startup, Main build, Evaluation (categorisation taken from Greene (1996)), and for each of the three main parties involved: project manager, project team, customer (categorisation taken from DeMarco & Lister, 1987), we have designed a separate list of questions, equalling $3 \times 3 = 9$ questionnaires. Furthermore, we have designed a project household questionnaire with some general data like staffing, billing, milestones, cost, effort, size, etc. Each questionnaire contains questions only for the specified role to be interviewed, during the specified phase. Guidelines for the construction of the latest version of the questionnaire were taken from the work by Oppenheim (1990).

From the results of the interviewing sessions, it became clear that we needed to capture especially the following problem categories:

- **projects data and experience**: billing, staffing, milestones, effort, cost, defects, management problems, communication problems, meetings, customer participation, technical infrastructure, working conditions, etc.;
- **products data and experience**: type, concepts, size, target audience descriptions, number of users, benefit for customer, media mix, platform, documentation;
- **critical success factors**: financial/work issues, risk analysis, general reference;
- **knowledge numbers**: e.g. boundary conditions, influences and best-before-date;
- **tips and tricks**: for example about bugs in multimedia development tools.

We used this second version of the questionnaires in a second iteration of another fourteen interviewing sessions. In tables 2.1 - 2.3, we give a quantitative description of the sessions of the second iteration.

**Table 2.1** Interview data for eight products (second set of interview sessions)

<table>
<thead>
<tr>
<th>Product type:</th>
<th>Duration (weeks)</th>
<th>Average staff</th>
<th>#</th>
<th># interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB Training</td>
<td>21</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>EPSS</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Intranet</td>
<td>50</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>EPSS</td>
<td>16</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CB Training</td>
<td>38</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CB Training</td>
<td>47</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CB Training</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CB Training</td>
<td>16</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

(The data from the projects that were analysed for which no interviews were held, was taken from extensive archival documentation.)

**Table 2.2** Interviews held for each of the project member roles

<table>
<thead>
<tr>
<th>Role</th>
<th># int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project managers</td>
<td>4</td>
</tr>
<tr>
<td>Designers/content experts</td>
<td>4</td>
</tr>
<tr>
<td>Graphic artists</td>
<td>3</td>
</tr>
<tr>
<td>Programmers</td>
<td>3</td>
</tr>
<tr>
<td>Testers</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 2.3 Project phases in which the interviews were held

<table>
<thead>
<tr>
<th>Phase</th>
<th># int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>planning phase</td>
<td>4</td>
</tr>
<tr>
<td>design &amp; main build phase:</td>
<td>6</td>
</tr>
<tr>
<td>evaluation phase</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3 The expert level of the multimedia project members that have been interviewed

<table>
<thead>
<tr>
<th>Level of experience</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not experienced (novice)</td>
<td>0</td>
</tr>
<tr>
<td>Slightly experienced (1 year)</td>
<td>2</td>
</tr>
<tr>
<td>Fairly experienced (&gt;3 years)</td>
<td>9</td>
</tr>
<tr>
<td>Experienced (&gt;5 years)</td>
<td>12</td>
</tr>
<tr>
<td>Highly experienced (&gt;10 years)</td>
<td>2</td>
</tr>
</tbody>
</table>

In all interviewing sessions, the following method was used:
1. Interviewer determines role of employee to be interviewed.
2. Interviewer determines the current phase in which the project is.
3. Interviewer and subject sit opposite each other in a closed room. Interviewer explains the nature of the interview and the eventual purpose of storage of the results.
4. Interviewer asks questions to subject; sometimes skips a question if it has been answered by a previous subject for the same project for the same phase.
5. Interviewer enters answers by the subject in digital form. Names are anonymised.
6. Digital results of interview are sent back to subject for final check.
7. After receiving final sign-off, the interviewer stores the results of the interview in the experience database.

We found that the most important multimedia specific problems can be summarised as follows (please compare to figure 1):

- **Organisational**: get the various disciplines to effectively co-operate; plan the various tasks for the different media to be realised; use knowledge numbers to do this;
• **Communicational:** listen to the perspective of other disciplines (especially important for the HCI expert); designing what needs to be communicated with the product;

• **Technological:** unstable tools, rapid changes in the software market (development tools), rapid changes in available technologies;

• **Managing user's expectations:** customers often are not able to estimate the technical feasibility/difficulty of their wishes and do not realise what a multimedia product approximately costs.

From the interviewing results, we designed a simple entity-relationship diagram as the basis for the multimedia experience database.

![Entity-Relationship diagram for the multimedia experience database](image)

**Figure 3** Entity-Relationship diagram for the multimedia experience database

This diagram has resulted in fifteen physical tables, plus eleven tables listing the various enumerations that people can choose from, such as billing, project phases, product types, target audience, codes size units, etc. The advantage of such predefined enumerations is that experience entries can later be more easily compared (for example find all projects that have resulted in the same product type).

The available tables are:

- product
- product size
- customer
- tips and tricks
- product experience
- project
- project problems
- project experience
- team
- defects
- cost
- effort
- mm experience descr.
- milestones
- person

The quantitative experience about communication, meetings, customer participation, etcetera, is stored into the project problems table, while the success factors and knowledge numbers are stored in the project experience table.
We wanted to offer the multimedia project team members and project managers the possibility of querying the database online, through a web browser, so that they could learn from the experience entered by others during the interviews. The database was in Access format, and we used Frontpage to set up a web site on a Windows-NT 4.0 server with the IIS/3.0 web server. Later we constructed a more advanced, open query in which various parameters as well as a freely choosable keyword could be entered, shown in figure 4. In this search, one can ask for project experiences relevant to a project manager, about communication, or meetings, technical infrastructure, etc., for a particular project phase, with a search keyword.

4. DISCUSSION

Here are some samples from the project interviews, taken directly from the experience database:

- "Problems with keeping the project organised? hm, firstly, there were a great many parties each with their own interests, resulting in white noise in communication and thus producing inefficiency in the project process. Secondly, there was no dedicated content-material expert on the project. Thirdly, there was a large geographical distance between the production team and the customer's headquarters."

- "It would have been better if there had been more formal communication; the value of that decreased because there was little formal communication anymore."
But the schedule was so tight that there was barely any time left for formal communication"

- "The office space in itself was okay, but we were there together with the customer, which resulted in two groups in one room. It wasn't bothersome, but we lacked spontaneity a bit.”.

- "The customer sticks firmly to their own opinion. And sometimes they really do have good arguments, and then it's difficult to convince them of something we think is better. Really difficult, I mean.”

- Numbers of estimated overhead per week on: insufficient communication between disciplines (4), insufficient technical infrastructure (8), insufficient HCI expertise available (2), lack of one person with several disciplines (2), and poor concepts and content (2).

One notable point is that all 25 multimedia experts are very enthusiastic about the interviewing sessions; they were all willing to take the time for the interview. Some of them remarked that even by just talking and actively thinking about these problems, you gain more insight into what went wrong. When we first showed them the experience database, all convinced that it is a crucial step in lifting the multimedia project process to a higher level. Of course, the usefulness of the experience database largely depends on the quality of the content.

5. CONCLUSION

The most important project problems that are specific to multimedia are:

- Get the various disciplines to effectively co-operate; plan the various tasks for the different media to be realised; use knowledge numbers to do this (organisational);

- Listen to the perspective of other disciplines (especially important for the HCI expert); designing what needs to be communicated (communicational);

- Unstable tools, rapid changes in the software market (development tools), rapid changes in available technologies (technological);

- Customers often are not able to estimate technical feasibility/difficulty, and do not realise what a multimedia product costs (managing user’s expectations).

These problems can partly be solved by offering a multimedia experience database, where multimedia experts can learn from experiences of the past. The current set-up of the multimedia experience database, in combination with the multimedia project questionnaires, forms a solid basis for realising the cycle described in this paper for improving control over multimedia projects.
6. FUTURE RESEARCH

A problem with the sharing of all this knowledge is that the multimedia experience has been delivered by over thirty multimedia experts from one and the same company. The information is therefore seen as somewhat 'company confidential', and has been restricted to within the firewall of the company (about 16,000 people worldwide). The database should be accessible anywhere on the world wide web. This functionality is potentially there, but needs to be unlocked by the firewall.

Since many of the experience that is stored has a qualitative character, it is hard to compare data entries and set up a way in which these properties can be objectively measured and improved. However, many of the qualitative experiences have been categorised into various subjects, and this in itself already provides a useful 'quantization'. The main goal of the further research is to investigate ways to set up a (quantitative) metric that makes use of the qualitative experience data, and thus builds up an extensive record of why things went wrong in multimedia projects, and how we can avoid these errors in later projects.

More about this research project can be found at http://is.twi.tudelft.nl/~jwvaJ, section ICOM (Improving Control Over Multimedia projects).

7. ACKNOWLEDGMENTS

We would like to express our sincere thanks to the following people that are directly involved in the ICOM research project: Toon Witkam, research coordinator at Origin; Johan Vader, Marcel Theunissen and Johan Versendaal, of Origin; prof. Jan Dietz at Delft University of Technology; and Jenny Preece and Tom Carey, for helpful comments in the course of the ICOM projects.

8. REFERENCES


9. BIOGRAPHY

Jan-Willem van Aalst works as a multimedia interaction designer at Origin IT Services in The Netherlands. He is doing a part-time research project on improving control over large multimedia projects, using a multimedia project experience database and a project metric with which to measure multimedia projects.

Charles van der Mast is a senior researcher and associate professor at Delft University of Technology. His research topics include human-computer interaction, user interface design, computer based training, and virtual reality. He has published his Ph.D. thesis about developing educational software in 1995.
BUILD-IT: A brick-based, integral solution supporting multidisciplinary design tasks

Morten Fjeld¹, Kristina Lauche², Stefan Dierssen³, Martin Bichsel³ & Matthias Rauterberg⁴

¹Institute for Hygiene and Applied Physiology (IHA), ETH Clausiusstr. 25, CH-8092 Zurich, SWITZERLAND
phone: +41-1-6323983, fax: +41-1-6321173
e-mail: fjeld@iha.bepr.ethz.ch

²Institute for Work and Organisational Psychology (IfAP), ETH Nelkenstr. 11, CH-8092 Zurich, SWITZERLAND
phone: +41-1-6326142, fax: +41-1-6321186
e-mail: lauche@ifap.bepr.ethz.ch

³Institute for Design and Construction Methods (IKB), ETH Tannenstr. 3, CH-8092 Zurich, SWITZERLAND
phone: +41-1-6322429, fax: +41-1-6321181
e-mail: {dierssen, mbichsel}@ikb.mavt.ethz.ch

⁴Center for Research on User-System Interaction (IPO), TUE NL-5612 AZ Eindhoven, NETHERLANDS
phone: +31-40-2475242, fax: +31-40-31930
e-mail: rauterberg@ipo.tue.nl

Abstract
BUILD-IT is an up-and-running system putting at work highly intuitive, video-based interaction technology to support complex planning and configuration tasks. It makes state-of-the-art computing and visualisation available to all kinds of users, without requiring any special computer literacy. Based on real, tangible
bricks as interaction handler, BUILD-IT represents a novelty to Human-Computer Interaction. With this tool, object manipulation and image display take place within the very same working area. Hence, new dimensions of prehension and direct response have been added to Human-Computer Interaction. Technology has a back-stage position, whereas creativity and human communication within multi-disciplinary expert teams is encouraged.

**Keywords**

Brick-based interaction, tangible objects, intuitive planning and design

1 INTRODUCTION

The concept of BUILD-IT is based on highly intuitive, video-based interaction technology, supporting complex planning and configuration tasks. It allows a group of people to sit around a table and handle projected objects with real tangible bricks as the interaction handler. Computer Aided Design (CAD)-based objects are manipulated and displayed within the very same working area. Hence, a new dimension of prehension and direct response has been added to Human-Computer Interaction (HCI). Dynamically coupled with an image displayed on the table, a perspective view of the planning situation is projected on a screen. This system makes state-of-the-art computing and visualisation available to all kinds of users, without requiring any special computer literacy. Instead of dominating cognitive and social planning processes, the system actually supports creativity and human communication.

For most planning tasks in systems engineering and architecture, drawings and 2D models have been replaced by CAD applications. This change has brought about a range of supportive tools for drawing and information processing. However, it also implies less immediate contact among CAD users, planning experts and sales people.

We began our work by performing a task analysis with potential user groups for our system. We observed that they spent a great deal of time in discussions with their clients and noticed that off-line CAD support is hardly available during sales trips. This lack of support sometimes caused misunderstandings with the designers 'at home', trying to communicate their solutions to the travelling sales people. Also, some of the customers were not familiar with 2D layout techniques; they were unable to imagine what an object would look like in 3D. Therefore, an easy-to-handle, 3D-planning tool proved to be of high interest to planning experts and to sales people. A distributed, networked system would additionally allow for interaction among users located at different sites.
Actually, modern management concepts like *Simultaneous Engineering* are based on dynamic interaction among co-operating experts. In this context, a tool should encourage team co-operation rather than each person working in front of a separate screen. Such needs can hardly be met by existing technologies like video conferencing. An adequate solution has to offer more intuitive, natural interaction.

All these considerations were taken into account in the design process of BUILD-IT. A system bringing support to early offering and design processes was the result. This tool is not intended as an alternative, but rather as a complementary aid for CAD systems. It allows for ready-made applications in various field, such as machine configuration, city and urban planning, architecture and interior design.

Tangible bricks represent a new way of interaction. Among others, this type of interaction was described by Ishii and Ullmer (1997), Underkoffler and Ishii (1998) and Fjeld, Bichsel and Rauterberg (1998). Rauterberg, Mauch and Stebler (1996) showed that a brick based interface is significantly easier and more intuitive to use than mouse and screen based interfaces.

Compared with physical, model-based layout systems, BUILD-IT additionally offers handling of CAD-objects and data management. It features cheaper, quicker and more exact object representation. The potential of computer mediated work is made readily available through automatic calculation of prices and time-to-delivery. Two-way communication with external CAD systems is assured, whereas animation and simulation offer design support at an expert level.

Offering higher efficiency in human communication, the system enables designers to accomplish their job with less travelling in less time. Distributed, networked systems offer simultaneous interaction for users located at different sites. Networked systems also encourage spontaneous distributed interaction, going far beyond the traditional, computer-based concept of teamwork.

In this paper, we will first give a system description, followed by an in-depth presentation of how geometric and meta-data is used by the system. Finally, we describe some user experiences.
2 SYSTEM DESCRIPTION

In a first step, we have designed a partial Natural User Interface (NUI) instantiation (Figure 1), as described by Fjeld, Bichsel and Rauterberg (1998). *Partial* means that distributed communication between multiple systems has not yet been implemented. As the task context, we chose that of planning activities for plant design. A system, called BUILD-IT, was realised. This is an application that supports engineers in designing assembly lines and building plants.

Figure 1: BUILD-IT.

The system enables users, grouped around a table, to interact in a space of virtual and real world objects. On the screen, a *side* view (Figure 2) offers a perspective of the planning situation. In the *working area* (Figures 2 and 4) there is an *above* view (planning situation as seen from above), *height* view (a slice of the side

Figure 2: Typical task solving situation with BUILD-IT. Interaction and display take place in the working area, whereas a perspective is offered by the side view.
view, for object height adjustment) and menu (split into a left and right part, offering new objects and functions). In the above view, height view and menu, objects can be selected, positioned, rotated and fixed (Figure 3). Functions (objects with functionality, like virtual camera, scaling, save & print) are selected in the menu and can be used in the above view.

**Figure 3:** The basic steps for user manipulations with the interaction handler.

The basic principle of BUILD-IT is shown in Figure 3. Users select an object by putting the brick at the object positions. The object can be positioned, rotated and fixed by simple brick manipulation. Using a material brick, everyday motor patterns like grasping, moving, rotating and covering are activated. Throughout these steps, there is a strong connection between cognitive processing and observable behaviour. The system dynamically supports the user needs for goal setting, planning, action and control. Hence, complete regulation of the working cycle (Hacker, 1994) is assured. The cost of making a mistake is low, since all vital operations are reversible. So, epistemic and pragmatic action (Kirsh and Maglio, 1994) are equally encouraged. To allow two handed operation, the system supports multi-brick interaction. A second effect of multi-brick interaction is that several users can take part in a simultaneous design process. Altogether, the set of NUI guidelines formulated by Fjeld, Bichsel and Rauterberg (1998) have been met.
The application is designed to support providers of assembly lines and plants in the early design processes. Graphical display is based on the class library MET++ (Ackermann, 1996). The system can read and display arbitrary virtual 3D objects as seen in Figure 4. These objects are sent from a CAD system to BUILD-IT using Virtual Reality Modelling Language (VRML). Geometry is not the only aspect of product data. There is a growing need to interact in other dimensions, such as cost, configurations and variants. Therefore, the system has been engineered to send and receive numerous forms of meta-data.

Figure 4: The working area with *above* view (situated in the centre), *height* view (situated on top) and *menu* (split into a left and right part, situated accordingly). The menu contains objects (robots, tables, conveyor belts etc.) and functions (objects with functionality; virtual camera, scaling, save & print etc.).

Figure 5: Multiple bricks allow for two-handed interaction.
BUILD-IT currently features the following user (inter-) actions (Figure 2-5):

- Selection of a virtual object (e.g. a specific machine) in a virtual machine store by placing the interaction handler onto the projected image of the machine in the object menu.
- Positioning of a machine in the virtual plant by moving the interaction handler to the preferred position in the plant layout of the above view.
- Rotation of a machine by coupling machine and brick orientation.
- Fixing the machine by manually covering the surface of the interaction handler and then removing it.
- Re-selection of a machine by placing the interaction handler onto the specific machine in the above view.
- Removing the machine by moving it back into the object menu (the virtual machine store).
- Modification of object size and height by operators in the method menu applied on objects in the above view.
- Direct modification of object altitude in the height view.
- Automatic docking of two or more objects along predefined contact lines within the above view.
- Scrolling of above view, height view and menus.
- Modification of the perspective in the height and side views by a virtual camera in the above view. Numerous virtual cameras, each representing a distinct perspective, can be used at a time. The last one selected determines the current perspective.
- Saving of the working area contents by a method menu icon.
- Printing of the views, also offered by a method menu icon.
- Multi-brick and multi-person interaction. All the previous (inter-) actions can be simultaneously executed by any of the bricks at the table.
- Simulation mode, supported by a simulation software (AESOP GmbH, 1997), shows real-time manufacturing. Steel sheets can be followed as they pass through different processes, like laser welding, chemical baths and drilling.

3 WORKING WITH VRML DATA AND META-DATA

The BUILD-IT system understands two different 3D-CAD data formats: VRML data and meta-data. We will pay most attention to VRML data, because they describe the complete geometry and visual characteristics of an object.

Additionally, depending on the field of application, users also need auxiliary object information. First, if configuration cost of the currently handled object is of interest, product name and unit price may be required. Second, in the case of process simulation (e.g. welding of metal sheets), different objects (e.g. robot, welding- or cleaning machine) and their characteristics (e.g. machine type,
capacity, preparation-, processing- and welding time) are needed. In both cases, object specific numbers and figures, named meta-data, are required. Such information is treated as separate data structure(s), and stored as meta-information (.mif) files.

Data exchange between a 3D-CAD system and BUILD-IT can be handled in two different ways: i) by the CAD-connection, and ii) by the Product Data Management (PDM)-connection.

![Diagram](image)

**Figure 6:** Data flow between the 3D-CAD, the PDM and the BUILD-IT system is based on CAD-connection, PDM-connection, integration and parts list integration.

### 3.1 CAD-CONNECTION

The most direct connection between a 3D-CAD system and BUILD-IT is the CAD-connection, as shown in Figure 6. CAD users are presented with a list of all available objects and can select the geometric data required for their specific planning session. The selected geometric data is converted to VRML format and offered by the CAD system as world (.wrl) files. Using the CAD-connection, the selected geometric data is then sent as .wrl files to BUILD-IT. For each
".wrl" file, a ".mif" file is generated. A ".mif" file contains additional object information like unit price and simulation parameters.

A VRML based connection offers the important advantage of data compression, allowing for reduced information flow and less object complexity. This feature is just as vital to object handling in the Web as with the BUILD-IT system. Without data reduction, only high performance CAD systems would be able to deliver multiple 3D object within acceptable time.

However, conversion, i.e. data compression and complexity reduction, also induce serious limitations. Circles are displayed as multi-edge polyhedrons, preventing an exact geometric object interpretation. Users wishing to position one object along the tangent of a second, circular object, with millimetres precision, can no longer be supported. A further consequence of data conversion is that direct feedback from BUILD-IT to the CAD system cannot be offered. Such feedback is impossible, because exact volume and surface information gets lost through conversion, and the original parts of an object can no longer be reconstructed. For this very reason, bi-directional communication of geometric and configuration data is not possible with the direct CAD-connection.

To make the description of the CAD-connection complete, we mention that meta-data, in this case ".mif" files, are also communicated via this connection. Since meta-data is exclusively being used by the BUILD-IT system, no feedback is needed, so the one-way CAD-connection is fully sufficient in the context of meta-data.

3.2 PDM-CONNECTION AND INTEGRATION

A more elaborate way to connect BUILD-IT with CAD systems became possible with the arrival of PDM systems. PDM systems do not only manage geometric data, they also offer product information such as parts lists. Parts lists are normally managed by larger database systems. By complete integration (Figure 6) of the PDM and CAD systems, geometric data can be converted into VRML data without having to interact with the CAD system. Selected objects are actually converted into VRML and meta-data in one integrated operation, called transfer process (Figure 6). This process is similar to the VRML-converter and the meta-data file generator, put together.

There is one major difference between the CAD- and the PDM-connection. As soon as a PDM user selects an object, a pointer is set on the corresponding parts list. This pointer is stored in the object’s meta-data. Supported by such pointers, it is possible, at any time, to load original parts lists and geometric data from the PDM system and to display them with the CAD system.
Connecting PDM with 3D-CAD and BUILD-IT systems, opens up new possibilities, far beyond managing parts list and geometric data. The main advantage of this combination is the concept of parts list integration (Figure 6). Parts list integration means bi-directional communication between the PDM system and BUILD-IT. Henceforth, it is possible to harvest the full advantages of a BUILD-IT planning session.

BUILD-IT users can assemble objects without having to care about causing any harm to original parts lists or geometric data. As soon as a planning process is accomplished, BUILD-IT generates a co-ordinates list. The list is the final result of the planning session and describes all the assembled objects. Supported by the parts list integration, communicated via the PDM system, the CAD system can now access, integrate and display the result of the planning session. Object modification that took place during the BUILD-IT planning session has no effect beyond that session.

4 USER EXPERIENCES

The BUILD-IT system was tried out with managers and engineers from companies producing assembly lines and plants. These tests showed that the system is easy to handle, intuitive and enjoyable to use. Most people were able to assemble virtual plants after only a few minutes of introduction. Some typical user comments were: 1) "The concept phase is especially important in plant design since the customer must be involved in a direct manner. Often, partners using different languages sit at the same table. This novel interaction technique will be a means for completing this phase efficiently." 2) "This is a general improvement of the interface to the customer, in the offering phase as well as during the project, especially in simultaneous engineering projects." 3) "The use of this novel interaction technique will lead to simplification, acceleration and reduction of the iterative steps in the start-up and concept phase of a plant construction project".

For the development of specific scenarios for each type of design task, we carried out interviews. Our subjects were expert designers, and the aim was to elicit their planning strategies, in order to get hold of relevant interactive parameters for a planning session.

5 DISCUSSION AND FUTURE PERSPECTIVES

Apart from enriching human-computer interaction in a direct and simple way, BUILD-IT has three further advantages over other Virtual Reality (VR) systems.
First, BUILD-IT supports group interaction while other systems, such as immersive VR, are single-user. Second, with its mixture of virtual and real tools, BUILD-IT allows for mixed (real and virtual) interaction, whereas other systems either use a mouse (pointer) or purely virtual tools. Third, BUILD-IT encourages teamwork among real persons interacting with real objects. All topics will be subjects for HCI research in general, and for the further development of BUILD-IT into an industry standard product in particular.

Plans for further development of BUILD-IT has been divided into three stages:

• **Task analysis and interaction design:** This stage will explore various ways of interaction, considering the task to be performed. It also includes preliminary experiments for cost/benefit studies of various forms of implementation, e.g. computer performance vs. group symbioses and user interaction. By the end of this stage, after approximately one year, various configurations of a functional BUILD-IT system, consisting of hardware and software, should be available.

• **User evaluation:** The second stage will consist of comparing the various configurations through usability studies. The aim is to investigate the relative advantage of different configurations relative to the task performed and to investigate the advantage of BUILD-IT vs. conventional desk-top systems, also relative to the task performed.

• **Prototyping:** Throughout the two first stages, the realisation can be at the level of wooden and wire solutions. The third stage will consist of developing these preliminary systems into a commercial product.

6 REFERENCES


BIOGRAPHIES

Morten Fjeld
1990
: MSc in Applied Mathematics, Norwegian Institute of Technology
1990-1997
: Design and realisation of real-time, industrial simulators, measuring systems and training equipment
1997-
: PhD student and research assistant in Cognitive Sciences and Human-Computer Interaction at ETH Zurich

Kristina Lauche
1995
: MSc in Psychology, Free University Berlin
1995-
: PhD student in Psychology
1997-
: Research assistant in Psychology at ETH Zurich

Stefan Dierssen
1997
: MSc in Mechanical Engineering, Technical University Clausthal
1997-
: PhD student and research assistant in Product Data Management at ETH Zurich

Martin Bichsel
1987
: MSc in Physics, ETH Zurich
1991
: PhD in Physics, ETH Zurich
1991-1992
: Postdoctoral fellowship at Media Laboratory, MIT Boston
1992-1996
: Senior lecturer in Computer Science at University of Zurich
1997-
: Senior lecturer in Computer Vision and Graphics at ETH Zurich

Matthias Rauterberg
1981
: MSc in Psychology, University of Hamburg
1985
: MSc in Computer Science, University of Hamburg
1995
: PhD in Computer Science, ETH Zurich
1995-1998
: Senior lecturer in Computer Science and Industrial Engineering at ETH Zurich
1998-
: Professor in Human Communication Technology at the Center for Research on User-System Interaction (IPO), TU Eindhoven
Combining Alternatives in the Multimedia Presentation of Decision Support Information for Real-Time Control

G. Herzog, E. Andre, S. Baldes, T. Rist
DFKI GmbH, German Research Center for Artificial Intelligence
Saarbrücken, Stuhlsatzenhausweg 3, Germany,
{herzog,andre,baldes,rist}@dfki.de

Abstract
Multimedia technology is emerging as a key element in the area of Decision Support Systems (DSS) since well-designed multimedia presentations help the human decision maker to assimilate relevant information more easily. The use of multiple media, however, increases the complexity of the presentation design task. Especially when complex information structures have to be presented under time pressure `ad hoc' solutions to presentation generation are getting more and more impractical, if not impossible to use. In this paper* we report on our approach to enhance a DSS for real-time traffic management with an advanced component for the automated generation of multimedia presentations. A common problem in this application class is the presentation of alternatives such as different explanations or predictions for a current traffic situation, or different sequences of control actions which may be initiated to resolve a problem. We describe a novel approach to provide aggregated information presentations rather than presenting alternatives just one after the other.

Keywords
Intelligent multimedia presentation, real-time decision support user interface, aggregated information presentation, automated multimedia presentation design

1 INTRODUCTION

Decision support systems (DSS) are interactive computer-based information systems that are designed to help human decision makers in utilising data and models in order to identify, structure, and solve semi-structured or unstructured problems and make choices among alternatives. Multimedia technology is emerging as a key element for

*The work described in this paper was partly supported by the project FLUIDS which is funded under the Telematics Application Programme by the European Commission, and partly by the AiA project funded by the German Ministry for Education and Research.
the adequate presentation of the complex information managed by a DSS since the ultimate goal is to effectively provide the human decision maker with the relevant information on the basis of available underlying data.

Especially in the area of real-time control applications there is a growing need to improve user-system interaction through multimedia-based decision support which integrates sophisticated problem solving capabilities with enhanced information presentation functionality. Potential application fields include for example: transport telematics for traffic control and traffic management, real-time control systems in industrial environments, monitoring and management of telecommunication networks as well as networks for power transmission and distribution, mission control and emergency management, and sophisticated decision support systems in the field of medical engineering.

The European project FLUIDS (Future Lines of User Interface Decision Support) aims at the design of a general environment for building intelligent interfaces to automated control systems that provide human operators with multimedia-enhanced real-time decision support. The integration of an advanced component for the automated generation of multimedia presentations constitutes a core element of the FLUIDS approach. In this paper, we report on the experience gained from adding this kind of multimedia functionality to concrete decision support applications in real-time traffic management. It turned out that one of the most challenging tasks is the adequate presentation of alternatives such as different explanations or varying predictions for a given traffic situation or several options for corrective control actions.

2 BACKGROUND

We are concerned with the development of an intelligent multimedia interface as backend to a decision support system which itself sits on top of a real-time traffic management system (cf. Figure 1).

![Figure 1 Components of an advanced traffic management system.](image-url)

The FLUIDS approach is being tested on different real-time traffic management systems currently operating in the cities of Madrid and Turin. Both systems are connected with large networks of sensors delivering real-time data about the traffic state. Considering various types of problems, three distinct applications are under development. The TRYS system in Madrid aims at generating proposals for traffic control strategies according to actual traffic conditions. UTOPIA, the urban traffic control
component of the 5T system in Turin operates fully automated instead. In this con-
text, FLUIDS is supposed to aid the traffic engineer in the diagnosis of system per-
formance as well as the analysis of the causes of possible faults and to suggest pos-
sible traffic model improvements. The 5T system is an integrated control system for
public and private traffic management with several subcomponents. A third FLUIDS
application builds upon SIS, the 5T public transport management component, to pro-
vide operators with suggestions for suitable control actions to recover from service
irregularities.

Though the above mentioned traffic management systems are build on elaborated
models of the domain and tasks both systems lack of sophisticated explanation ca-
pabilities such as to aid users in understanding how and why the system reaches
its conclusion, to convince users that conclusions drawn by the system are sound
and reasonable, and also assist in debugging the knowledge and problem solving
behaviour of the system. As a prerequisite to achieve these abilities, a knowledge-
based module for problem solving (PSM) has been developed using the Knowledge
Structure Manager environment (KSM, cf. (Cuena, Hernández & Molina 1997)).
This component includes qualitative models of the algorithmic processes of the un-
derlying traffic control systems, and is able to provide qualitative explanations of
proposed solutions for trouble shooting. As shown in Figure 1 the PSM component
is also connected to the user interface. On request by the user, it provides three types
of information: (1) the current situation, i.e. `What happens?', (2) forecasts, i.e.
`What may happen?', and (3) potential control actions to be initiated for trouble
shooting, i.e. `What to do?'.

A typical task of a traffic operator is to recognise the most critical network link,
to identify the potential causes of an abnormal situation (e.g., by comparing all the
estimated parameters with the nominal and historical parameters), and to select an
applicable control action to solve the problem. For example, in response to the ques-
tion `What is happening on the network?' the system will present one or more areas
where the difference between estimated delay of a bus line and the tolerable delay
exceeds predefined thresholds (for example, an absolute threshold of 150 seconds).
Concerning the follow-up question `What to do?' , the system will then inform the
operator about possible control actions for solving the identified problem. Needless
to say that it is the task of the user interface to present such information to the user
in a way that effectively supports the operator in time-critical decision making.

The initial versions of the traffic management components within both systems,
TRYS and 5T, are equipped with window-based interfaces. All these interfaces em-
ploy different media for the presentation of information; full text, short messages
below sentence level, maps, and abstract diagrams, e.g., a horizontal bar with mark-
ers on it as an encoding of a bus route with stops. Our evaluation of the informa-
tion presentations delivered by the interfaces, however, revealed a number of serious
shortcomings:

• poor temporal output coordination, especially when distributed on different win-
dows;
• no follow-up questions on presentations because of lacking semantic representations of system output;
• no means to condense presentations in order to reduce both redundancy and presentation time;
• little flexibility in the system's presentation behaviour because of a 'hardwired' mapping from data instances to presentation instances.

Further requirements for an improved system were obtained directly from potential system users. The interviewed users were experienced operators in the system control centres at Turin and Madrid. As expected, there was almost no need to generate a broad variety of different presentations to accommodate for different user profiles. Moreover, the operators indicated a strong preference for having only a limited number of presentation patterns with which they could easily get familiarised. For example, the operators preferred a small number of display frames with a fixed layout for graphics and text output, a small number of different graphic types (overview maps, network diagrams, line charts). On the other hand, there was a strong demand for improving the system's presentation capabilities by means of aggregation mechanisms. The less an operator had to browse through lists of textual messages and to switch between display frames in order to perform a supervision task or to decide among potential control actions, the better the system.

3 PRESENTATION TASKS AND PRESENTATION TYPES IN THE TRAFFIC MANAGEMENT APPLICATION

The task of presenting information is usually conceptualised as a mapping from given information units (domain concepts) to presentation instances (media objects or combinations of media objects). Following this view, we have to identify and classify the concepts relevant to the underlying domain, potential presentation instances, and the conditions under which a certain presentation instance should be chosen.

3.1 Domain concepts and their representation

As mentioned in section 2, the domain knowledge is modelled and represented within the KSM framework for the development and maintenance of large and complex knowledge-based applications. For the purpose of this paper we restrict ourselves to briefly introducing domain concepts which are referred to in other parts of the paper. These concepts are locations, vehicles, streets, routes, states, events and situations, and control actions.

Locations and trajectories of moving objects are conceptualised as particular positions or regions over a background frame. The background frame may be a geometric map of a town or neighbourhood so that all represented locations have denotations in the real world. However, a background frame may also be an abstract graph structure (e.g., providing topological information on routes). Domain objects are vehicles,
streets, routes, bus lines, traffic signs, etc. Each represented object is internally accessible through a unique identifier, and may have a number of attributes assigned to it (e.g., a location, a "pretty name" or an icon for its graphical display). As some attributes of domain objects may change over time, object descriptions may vary from one instance in time to another. States and events are described by means of predicates that may hold for an object or some objects at a certain instance in time or over a certain time period. For example, a bus may be operable or broken, a bus line may be delayed, whereas a conjunction event may have been recognised or forecasted by the system. Situations are introduced to characterise relevant aspects of complex traffic situations. Situation descriptions may comprise a number of state and event descriptions For example, the Lisp-style representation below captures the situation where a bus-line is delayed due to the delay of a bus (vehicle bus#5 has a delay of 17 minutes).

\[
\text{(current_situation \ (vehicle_state \ bus#5 \ delayed) )} \\
\text{(vehicle_location \ bus#5 \ loc#188) } \\
\text{(vehicle_delay \ bus#5 \ 17) }
\]

Explanations are event sequences whose outcome would be consistent with the current situation. For example, if a traffic problem has occurred, the operator may be interested in the events which caused the problem. In some cases, several plausible explanations may be found due to the system's incomplete knowledge of the real world. Predictions are possible future traffic situations. Starting from the current situation, they are computed by the problem solving module, e.g., through a traffic simulation process. In some cases, a high degree of uncertainty may lead to several potential situations of the same likelihood. Control actions are actions which may be initiated in order to resolve a traffic problem. For example, if a bus breaks down at a certain location, the diagnosis system may suggest either to send a replacement bus which continues the service, or if feasible, to make the passengers wait for the next bus of the same line.

In the following, we introduce a simplified notation for actions, action sequences and alternatives. Actions are characterised by an action type and a list of action parameters in the underlying domain representation. Furthermore, an action can be either primitive or a composition of other actions. Action terms are inductively defined over the set of primitive domain actions:

1. each primitive domain action is an action term;
2. if \(a_1, ..., a_n\) are action terms, then the action sequence of the form \([a_1; ...; a_n]\) denotes the temporally ordered sequence of the actions \(a_1, ..., a_n\) and is also an action term;
3. if \(a_1, ..., a_n\) are action terms, then the list of alternatives has the form \((\text{Alt } a_1, ..., a_n)\) and is also an action term. In case of control actions, it refers to a list of several actions from which the operator has to choose exactly one.
4. Actions which are described by action terms may have a hierarchical structure including alternatives since each \(a_i\) in a sequence or a list of alternatives by itself is an action term.
3.2 Presentation types

For the traffic management domain, we have to define presentation types for accomplishing tasks such as presenting:

- objects, attributes and states of objects, object locations and trajectories;
- relevant aspects of complex traffic situations, such as events and involved objects;
- explanations, i.e. causes for the occurrence of an event or a problematic traffic situation;
- predictions how a certain traffic situation may evolve (e.g., within the next hour);
- sets of potential control actions from which the operator has to select one or more in order to avoid or resolve problems.

![Figure 2 Graphical display types of the Fluids demonstrator: Street network and bus line diagram.](image)

Two different sample displays are shown in Figure 2. In accordance with the user requirements study, the presentation media text, speech, 2D graphics and 2D animation are supported in the combinations listed in Table 1. In case of language (text or speech) predefined sentence patterns are used to encode descriptions for object states, events and actions. Because the operators preferred to see a kind of textual record, the use of the medium speech is supported only on demand and always in addition to text. Static graphics include several types of map displays, and special purpose diagrams such as bus line visualisations. Basic domain objects such as vehicles are graphically represented by icons. The set of icons comprises also conventionalised icons for the indication of some events (e.g. accident) and actions (e.g. driver exchange), and a few marker icons (e.g., blinking circles and arrows) which are used to draw the viewer's attention to a certain location on the display. For animations we distinguish between visualisations of moving objects on a map background, and the temporally coordinated annotation of a static display. That is, starting with a background display, the final static display is completed step by step with annotations before the operators eyes. In contrast to the usual form of animation, this type of an-
imation has the advantage that the last image frame can be viewed stand-alone as a static graphics which encodes all the relevant information that has been added during the preceding animation.

<table>
<thead>
<tr>
<th>Current Situation</th>
<th>text</th>
<th>sentence by sentence enumeration of the occurred events</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speech</td>
<td>additional spoken telegraph-style descriptions (optional)</td>
</tr>
<tr>
<td></td>
<td>static graphics</td>
<td>annotated maps (showing locations of involved objects)</td>
</tr>
<tr>
<td></td>
<td>animation</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Explanation</th>
<th>text</th>
<th>sentence by sentence enumeration of the potential causes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speech</td>
<td>additional spoken telegraph-style descriptions (optional)</td>
</tr>
<tr>
<td></td>
<td>static graphics</td>
<td>visualisation of diagnosis results: (a) static graphics: dynamic data such as object trajectories are shown by arrows;</td>
</tr>
<tr>
<td></td>
<td>animation</td>
<td>(b) animation: dynamic scenes can be played back and forth with arbitrary speed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prediction</th>
<th>text</th>
<th>sentence by sentence enumeration of the predicted events</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speech</td>
<td>additional spoken telegraph-style descriptions (optional)</td>
</tr>
<tr>
<td></td>
<td>static graphics</td>
<td>visualisation of simulation results (similar to explanation)</td>
</tr>
<tr>
<td></td>
<td>&amp; animation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control Actions</th>
<th>text</th>
<th>sentence by sentence enumeration of the proposed actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>speech</td>
<td>additional spoken telegraph-style descriptions (optional)</td>
</tr>
<tr>
<td></td>
<td>static graphics</td>
<td>static graphics are modified by subsequent annotations on a</td>
</tr>
<tr>
<td></td>
<td>&amp; animation</td>
<td>map to show locations and trajectories of involved objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and locations where actions take place</td>
</tr>
</tbody>
</table>

**Table 1** Presentation types used in FLUIDS to convey domain information.

### 3.3 Presentation planning

To map information units onto multimedia presentation instances, we rely on our framework for the representation and generation of multimedia presentations (cf. (André, Finkler, Graf, Rist, Schauder & Wahlster 1993, Rist, André & Müller 1997)). In this framework, we operationalise the generation of multimedia presentation by means of a goal-driven, top-down planning mechanism. The presentation planner receives as input a communicative goal (for instance, the user should be able to localise the malfunctioning vehicle on the network) and a set of generation parameters, such as target group, presentation objective, resource limitations, and target language. The task of the component is to select parts of a knowledge base and to transform them into a multimedia presentation structure. Whereas the root node of such a presentation structure corresponds to a more or less complex communicative goal, such as describing a prediction for a traffic situation, the leaf nodes are elementary generation or presentation acts, currently for text, graphics, and animations.
In order to cope with the dynamic nature of most multimedia presentations, the presentation planner has been combined with a temporal reasoner based on (Kautz & Ladkin 1991) whose task is to determine a preliminary presentation schedule. Since the temporal behaviour of presentation acts may be unpredictable at design time, the schedule will be refined at presentation runtime by adding new temporal constraints to the constraint network.

We use so-called presentation strategies to represent knowledge concerning how to decompose a given presentation task into subtasks or, in case of elementary subtasks, which media objects should be used to convey the subtasks. Presentation strategies consist of a header, a set of applicability conditions, a collection of inferior acts, a list of qualitative and metric temporal constraints, and a start and an end interval. The header of a strategy corresponds to a complex presentation act such as presenting a traffic situation. The applicability conditions specify when a strategy may be used and constrain the variables to be instantiated. The inferior acts provide a decomposition of the header into more elementary presentation acts. Qualitative temporal constraints are represented in an 'Allen-style' fashion which allows for the specification of thirteen temporal relationships between two named intervals: before, meets, overlaps, during, starts, finishes, equal and inverses of the first six relationships (cf. (Allen 1983)). Allen's representation also permits the expression of disjunctions, such as (A (before after) B), which means that A occurs before or after B. Metric constraints appear as difference (in)equalities on the endpoints of named intervals. They can constrain the duration of an interval (e.g., (10 <= Duration A2 <= 40)), the elapsed time between intervals (e.g., (4 < End A1 - Start A2 < 6)) and the endpoints of an interval (e.g., (Start A2 >= 6)).

The basic repertoire of presentation strategies for the traffic management application has been defined in a straightforward manner. For each of the information types listed in section 3.1 at least one presentation strategy has been defined. An example of a presentation strategy is shown below. It may be applied to inform the operator about a delay of a vehicle (e.g. a bus) via graphical and textual means.

```
(define-strategy
 :HEADER
 (A0 (INFORM-DELAY-DETAILED P A
       ?v-location ?v-delay ?delay-label))
 :INFERIORS
  (A2 (VERBALIZE-VEHICLE P A ?text-window ?vehicle))
  (A3 (SHOW-RED-BLINKER P A ?graphic-window ?v-location ?pos-1))
  (A4 (VERBALIZE-VEHICLE-DELAY P A ?text-window ?minutes))
 :QUALITATIVE ((A1 (e) A2) (A2 (s) A3) (A2 (m) A4) (A4 (m) A5))
 :METRIC ((20 <= DURATION A3 <= 30) :START A3 :FINISH A3)
```

At this stage of the project, two improvements over the original interfaces of the traffic management systems (TRYS and 5T) have been achieved. It is now possible
to ensure a proper temporal coordination between presentation acts, only by specifying temporal relationships between the inferior acts in the strategies. Furthermore, there is now a clear separation between the representation of domain knowledge and presentation knowledge which facilitates the modification and fine tuning of presentation types. However, the basic repertoire of presentation strategies defined so far did not help to avoid redundancies when presenting event and action sequences with overlapping subparts. This problem occurs when alternatives have to be presented, e.g., in situations in which the system comes up with different explanations or predictions for a certain situation, or with different sequences of control actions for problem solving. In case of the FLUIDS system, usually a single explanation and a single prediction is delivered but for control actions the set of alternatives does frequently contain 2–3 instances. In order to further improve the system's presentation abilities, the aggregation task has to be addressed, too. In the following section, we concentrate on control actions and sketch how our approach handles aggregation tasks.

4 AGGREGATED PRESENTATION OF CONTROL ACTIONS

To illustrate the problem, let's consider the following scenario: The system has informed the operator that a bus, say bus#11, broke down at location loc#347 and is now no longer able to continue its service for the corresponding bus line. After the operator has asked for advice on what to do, the diagnosis subsystem suggests two alternative action sequences which may be initiated to fix the problem.

The first solution is to send a repair car and a replacement bus to the location where the broken bus#11 is standing. Then the drivers are exchanged and the passengers will be transferred to the replacement vehicle. Finally the broken bus will be towed away with the repair car. The first action of the second solution coincides with the first action of the first alternative. That is a repair car is moved to the location of bus#11. However, instead of using a replacement bus, the system suggests to wait for the arrival of the next bus of the same line. Then the passengers have to change to bus#12 and the broken bus will be towed away. Using a Lisp-style notation, the output of the diagnosis component is as follows:

```
(Alt [ (move repair-car#5 loc#347);
   (move bus#15 loc#347);
   (exchange-drivers bus#11 bus#15 loc#347);
   (transfer-passengers bus#11 bus#15 loc#347);
   (tow-away bus#11 repair-car#5 loc#347) ],
[ (move repair-car#5 loc#347);
   (wait-for-next-bus-of-line bus#12 loc#347);
   (transfer-passengers bus#11 bus#12 loc#347);
   (tow-away bus#11 repair-car#5 loc#347) ] )
```
4.1 Subsequent presentation of all alternatives

A straightforward way of presenting potential control actions is to produce first a kind of advance organiser which introduces the alternatives and second to describe all alternatives in detail. If we apply this strategy on the previous example, we get the presentation structure shown in Figure 3.

![Presentation Task and Corresponding Presentation Structure]

While it is easy to define a presentation strategy for this case, the resulting presentations are often long-winded and thus are not suitable for the support of decision-making under time pressure. This is especially crucial when speech and animation get involved in the descriptions of subactions since the total presentation time is determined by the sum of the time needed for each single description. Furthermore, such presentations make it very difficult for the decision maker to recognize similarities and differences between alternatives.

4.2 Factoring out common subactions

Obviously presentation time can be saved if it is possible to restructure the presentation in such a way that descriptions of common subactions only appear once in the presentation. The two sequences of the example have the subactions \((\text{move repair-car} \#2 \text{ to loc} \#347)\) and \((\text{tow-off repair-car} \#2 \text{ bus} \#11 \text{ from loc} \#347)\) in common. Our approach to factor out such common parts is to reformulate the given presentation task into a new task with a less redundant structure. Figure 4 illustrates the intended reformulation. In essence, we go through the list of control actions in order to figure out whether there are pairs of common actions. If such pairs exist, the given presentation task is reformulated into a new task which can be accomplished more efficiently than the original task. The rational behind this approach is the assumption, that we can use similar presentations for similar action instances. However, it is not
always advisable to perform all possible transformation because the resulting structure may become even more difficult to present as the original list of alternatives. In the FLUIDS system, we restrict ourselves just to factor out common start, middle or end subsequences and avoid structures with nested branchings.

**Figure 4** Presentation structure for the reformulated presentation task.

For the combined presentation of the two alternative control actions we deploy the graphical display shown in Figure 5. It is used to convey the trajectories of the involved vehicles. While both action sequences comprise the same trajectory for vehicle r-5, the trajectories of b-15 and b-12 represent alternatives.

**Figure 5** Combined graphical display of two alternative control actions.

### 4.3 Factoring out common aspects of actions

In some cases, the only difference between two alternatives is only due to different bindings of some action parameters. That is, two actions a and b are of the same type,
but at least one action parameter has a different binding. Consider for example the situation in which the operator should send a repair car to a certain location but may have the choose between a red and a blue car. The presentation of this alternative may be shortened by factoring out the common aspects of nearly similar actions, e.g. by saying `move the red or blue repair car to loc...'. This can be achieved by means of a further reformulation strategy which would merge

\[ \text{Present} \ (\text{Alt} \ [ \ldots \ (\text{move} \ \text{repair-car}#1 \ \text{loc}#347) \ \ldots] , \\
\ldots \ (\text{move} \ \text{repair-car}#2 \ \text{loc}#347) \ \ldots]) \]

into \( \text{Present} \ldots \ (\text{move} \ (\text{Alt} \ \text{repair-car}#1 \ \text{repair-car}#2) \ \text{loc}#347) \ \ldots) \).

Of course such reformulations make only sense if there is a presentation strategy which is able to handle the encoding of alternative parameter bindings. In the example presented above we have a slightly different case concerning the subaction \textit{transfer-passengers} which occur in both alternatives. The only difference on the propositional level lies in the binding of the second parameter which is bound to \text{bus}#15 in the first sequence, and to \text{bus}#12 in the second alternative. However, in this case the action context determines which of the two bindings must be chosen. If an aggregation strategy is applied, we have to ensure that this dependency is reflected on the surface level, too. Instead of just saying “transfer passengers from the broken bus (bus#11) to the substitute bus (bus#15) or the next bus in line (bus#12)”, we would mark the dependency by adding “respectively”. Unfortunately, it can be quite difficult to determine whether or not an alternative for a parameter binding depends on a previous decision. In the transfer-passenger example, it may suffice to trace back the occurrence of the corresponding parameters and to figure out that the two bindings (replacement bus bus#15 and next bus of line bus#15) were introduced in alternative preceding subsequences. In the general case, however, deeper reasoning on the domain knowledge will be required in order to avoid useless factoring.

### 4.4 Embedding the approach into presentation planning

This approach has been included into our presentation planning environment by augmenting the repertoire of presentation strategies by task-reformulation strategies. The header of such task-reformulation strategies represents the initial task while the body refers to its reformulation. The constraint slot of the strategies is used to specify conditions under which a reformulation should be performed. For example, a constraint for factoring out a certain subaction is that it must occur in two alternative sequences. Further constraints have to be formulated in order to avoid too many reformulations. For example, we avoid reformulations which lead to nested branching structures as they often become quite difficult to present. Whenever the planner encounters a new presentation task, it first tries to reformulate the task by using task-reformulation strategies before decomposing it by applying presentation strategies. Note that in case a reformulated task cannot be solved eventually, the planner will launch a backtracking process that withdraws the reformulation decision.
5 RELATED WORK

There are enormous efforts of the software industry to provide multimedia functionality with their DSS products. For example, many database vendors aid decision makers within a business context in accessing and presenting the information provided by an enterprise decision support systems. In this application area capabilities for information presentation range from simple tabular to advanced multidetail reports with all types of graphs and charts. Such systems incorporate dedicated generation modules such as table formatters or chart drawing components. Promising experiences in enhancing DSS with multimedia components have also been reported from research activities in the area of medical decision making. However, current DSS yet do not take advantage of more recent methods for the automated design of multimedia presentations (cf. (Feiner & McKeown 1991), (Maybury 1991), (Stock 1991), (André et al. 1993), (Arens, Hovy & van Mulken 1993), and (Roth & Hefley 1993) for an overview). Vice versa, real-time decision support is only rarely chosen as an application domain for automated presentation generation, like for example in (Sutcliffe & Faraday 1994). This may be one of the reasons why the aggregation problem has not been addressed very detailed so far in this research community.

With the application data on the one side, the generated presentation parts on the other side, and the presentation generator in between, there are three different approaches to information aggregation which aggregate either over (1) domain data, (2) media objects, or (3) intermediate presentation structures. Following the first approach means to introduce additional concepts in the representation of the domain and the definition of presentation strategies for these additional concepts. The problem with this approach is that it blurs the borderline between domain modelling and specification of presentation knowledge. In our project consortium the engineers responsible for modelling the domain didn't feel comfortable with the idea of defining new domain concepts "just" to improve the systems presentation abilities. They were in favour of keeping the modularization of tasks and responsibilities as it was in the initial systems. Approaches that relate to the second alternative can be found in the area of text summarisation (e.g. (Spärck-Jones, Endres-Niggemeyer, Hobbs, Liddy & Paris 1993)). In this community, a number of techniques have been developed in order to derive a summary from a source text. Such an approach seemed to inefficient for our application as we would have to generate first a complete presentation as input for a subsequent aggregation process. Approaches that fall under the third alternative have in common that they try to perform aggregations on representation formats that are used in the generation process. These formats can be media-independent presentation acts, presentation acts to be conveyed in a certain medium, or media-specific structures of presentation units, such as preverbal messages during text generation. Usually, an aggregation module is added between the content planner and the text generator (for example, see (Dalianis & Hovy 1993), (Shaw 1983)).

Our approach aims at aggregations at the level of presentation acts, too. However, we apply restructuring strategies at an early stage during presentation planning. This approach enables us to consider dependencies between content structuring and
aggregation which are more crucial in the FLUIDS application than dependencies between aggregation and realization since we rely on pre-stored text patterns and schema-based graphics instead of fully-fledged media design as for example the graphics design approach proposed in (Casner 1991).

6 CONCLUSION

In this paper we have reported on our work to equip an existing real-time traffic management application with a component for the automated design of multimedia presentations. In particular, we sketched how our framework for plan-based presentation design was adapted and augmented to suit this application. From the viewpoint of research on real-time decision support systems, this work may be of interest because it enables us to replace ad hoc solutions for the handling of crucial presentation issues by a principled approach for the intention-based coherent structuring of presentations and the temporal coordination of media items. On the other hand, real-time decision support appeared as a promising, but challenging application area for research on automated multimedia generation. To ensure that presentations are both short and easily to follow for time-pressured controllers, the generation of aggregated information presentations is an important issue which has to be addressed.

In our proposed solution a presentation planner attempts to reduce the number of propositions to be communicated by factoring out information units such as common actions of alternative action sequences. The approach helped to significantly improve the presentation abilities of the traffic management system in comparison to the original interface.

However, there is still much room for further improvements. First of all, it is important to extend the set of multimedia presentation types for condensed information presentations. While in the case of text valuable inspirations can be found in the literature, pioneering work is still required when it comes to graphics and animation. Currently, we are experimenting with graphical forms for the presentation of alternatives. For example, alternative object movements may be visualised through colour coding, or more dynamically, by alternating superimpositions of arrows for the alternative trajectories. In the current implementation, we are quite restrictive when factoring out common information units. We do not perform reformulations which would produce more complex branching structures. This restriction increases the chance that a suitable presentation can be generated for a reformulated task. On the other hand, there may still be unnecessary redundancy in generated presentations of alternatives. Another issue concerns the generality of our task reformulation strategies to aggregation. They essentially merge separated items (i.e. action sequences) in case that they share common parts (i.e. subactions). This approach was reasonable since in the FLUIDS context we had to start from a given domain representation, namely the one being used in the diagnosis system. One could certainly imagine a diagnosis system which delivers a graph-like structure instead of a list of alternatives. In this case, part of the aggregation task would be to split the graph structure into reasonable units which can be presented together.
REFERENCES


Visualization — the transformation of data and information into multimedia including pictures, animation and 3D scenes — enables users to understand information more naturally. It reveals patterns and relations in the information which may otherwise remain hidden. As a consequence, it can provide a single user with enough valuable information to support decision making. In addition to this, visualization can also be used to explain information to other people. In this case, the results of visualization are deployed as arguments in collaborative decision making.

This paper discusses a distributed visualization architecture which supports collaborative decision making. The architecture is designed with the following consideration in mind: “multiple users, with different information needs, require multiple views or perspectives of the data.” Additionally, in order to support the cooperation between users during the decision making process, we extend the architecture with collaborative aspects including session management, and the exchange of visualization perspectives.

1 INTRODUCTION

Visualization is used to give better insight into data by showing a visual representation of the information. Visualization is becoming increasingly important because people are suffering from an information overload caused by enormous amounts of
data. By using visualization we can first explore a comprehensive overview of the information and later decide to zoom in on the details.

Currently, people are using visual representations of information for two different purposes. First, visualization is often used to understand information. A visualization gives quick insight into information using humans' remarkable perceptual abilities (Shneiderman 1998). Second, visual representations are used to show information to other people. Shneiderman (1998, p. 522) states that the bandwidth of information presentation is potentially higher in the visual domain than it is for media reaching any of the other senses. For example, newspapers are full of graphs to show economic growth or the developments on the stock exchange market. In the first case, when using visualization to understand information, we often apply it individually (although it is surely useful to try to understand information in a group process). In the latter case we are communicating with other people because we try to illustrate something, or we want to convince them of our point of view.

In addition to static visualizations (2D images), current technology enables a new form of visualization: the interactive visualization of dynamic data. In recent years, the desktop computer has evolved from a text/picture based system to a fully multimedia-enabled workstation. This offers a great opportunity to deploy visualization on multimedia desktop computers. Visual representations consisting of interactive 2D or 3D animations enable the visualization of dynamic data coming from, for example, running simulations. Furthermore, multimedia PCs connected to fast networks allow desktop video conferencing, enabling direct user-to-user communication.

**Structure** The next section illustrates why visualization is useful to support collaborative* decision making in a business process re-design project. Section 3 briefly describes the *distributed visualization architecture* (DIVA), intended for multi-user visualization. After discussing issues in collaborative visualization in Section 4, we will describe our architecture extended with collaborative aspects in Section 5. The sixth section illustrates DIVA from a user's point of view by means of a sample visualization. Finally, in Section 7, we will end up with conclusions.

## 2 RE-DESIGNING BUSINESS PROCESSES AT THE GAK

Our example concerns the GAK (*Gemeenschappelijk Administratie Kantoor*), which is the largest provider of social security in the Netherlands. The GAK organization's main services are the registration and collection of insurance premiums, and the payment of social security benefits.

ASZ, which is the IT company of the GAK Group, builds and maintains the information systems that the GAK is using. Currently, the information infrastructure consists of several large databases, hundreds of separate applications and little inte-

* In this paper we will use the terms collaboration and cooperation interchangeably.
gration. To improve this, ASZ is investigating a software architecture that combines the databases, legacy software and new applications into a highly integrated system.

This development will certainly have an impact on the business processes at the GAK. The company will be able to improve current services and to offer new ones. However, deciding how the new business processes must be organized and what the consequences of some decisions will be is not at all trivial.

To assist the managers in studying the alternatives we create business-process simulations to execute the re-design alternatives (Eliëns, Niessink, Schönhage, van Ossenbruggen & Nash 1996). The managers are now able to run the simulations and experiment with the re-design alternatives themselves. To fully exploit the potential of business simulations we allow the managers to visualize and discuss both the results of the simulation, e.g. the costs and profits, and the running simulation itself, e.g. to illustrate the activities in the re-designed alternative.

**Example: registration of new companies**

As a concrete example, consider the process of registering the employees of a newly established company for social security. In the past, the employer had to go to a number of counters to fill in the required forms. When the client had forgotten something needed for the registration, he had to go back to get it and start the whole procedure again.

As a re-designed alternative we want to explore two options. First, all the paper forms could be combined into a single computer application. All forms could then be filled in at once, in dialogue with a single GAK employee. Second, a GAK employee might be able to go to the newly established company. There, using a laptop, she could fill in all of the needed information by asking it directly to the client on the spot.

To decide which alternative is preferable, we have to consider a number of aspects including the cost of the alternative, the satisfaction of the clients, and the time needed to register the company.

A visualization of the business process flow is useful in explaining the business process alternatives. This illustrates who is performing which tasks and how the information flows through the model. Additionally, a geographical visualization shows how far and how often clients and employees of the GAK have to travel. The costs, waiting times and other statistical information of the re-designed alternatives can be presented using statistical visualizations, such as charts and histograms.

The decision makers, who are spread out over the country, plan to make the definitive decision at a meeting. However, before that, they want to prepare and discuss several alternatives. The above mentioned visualizations offer the decision makers (and other interested employees) a common ground for discussion.

Essentially, we want to support two forms of collaboration: synchronous distributed and face to face (Ellis, Gibbs & Rein 1991). In order to help the participants prepare for the meeting, we first support synchronous distributed collaboration where the
users cooperate at the same time but in different places. Secondly, at the meeting, where the decisions are made, the decision makers will discuss the selected alternatives *face to face*, i.e. same time, same place.

### 3 MULTI-USER VISUALIZATION

As the above example illustrates, it is useful to take visualization from the single user domain into the realm of distributed multi-user systems. This makes it possible to discuss shared information sources.

However, multiple users with different backgrounds have different information requirements. In the example above some managers might be interested in the resource allocation (who is using what) of the re-design alternative, while others might be more interested in the financial aspects. To support these different information requirements, multiple perspectives (or views) on the information are required. So, based on the same simulation, we distinguish alternative perspectives that visualize different aspects of the re-designed process.

The need to have multiple perspectives was the main motivation for designing the *distributed visualization architecture* (DIVA). Additional requirements were the support for interactive visualization to allow for experimentation, and visualization at the user’s desktop by means of a networked or web-based architecture (Schönhage & Eliëns 1998).

We regard the process of visualization as a transition of data through a sequence of models, starting with the generation of data and ending with the presentation of a visualization (Schönhage & Eliëns 1997). To allow for multiple perspectives on the data, we introduce an intermediate model between the generation and presentation of information. This intermediate model contains information based on the originally generated data, adapted to the information requirements of its users.

![Conceptual architecture](image)

**Figure 1** Conceptual architecture

Figure 1 depicts our architecture on a conceptual level. The primary model is the source of the information and contains explicitly or implicitly all information available. A conceptual mapping gives us the ability to adapt the raw information in the primary model to our information needs. Consequently, information in the derived model differs from data in the primary model in two ways. Primarily, only valuable information is selected to be present in the derived model and, secondly, information derived from primary data is added in the derived model.
How we present the derived information is specified in the presentational mapping. Here, information concepts in the derived model are mapped onto generic visualization primitives. The final presentation is the content of the presentation model. For example, when using DIVA for 3D visualization, the presentation component contains a 3D scene through which end users can navigate. For a more detailed description of DIVA see Schönhage & Eliëns (1997) and (1998).

4 COLLABORATIVE VISUALIZATION

In DIVA, multiple users can have their own presentation model (perspective) while sharing a common resource. However, in this approach the different users have the feeling that they are the only user of the shared resource. There is not yet support that allows a user to be aware of other users or to interact with them. Our goal is to expand the architecture to support users to collaborate with each other. Here, 'to collaborate' means that the users are able to discuss visualized information in order to reach a decision.

The next section will address the issues that are involved in this restrictive notion of collaborative visualization. Then, we will discuss the requirements for an extended DIVA architecture in Section 4.2.

4.1 Issues

DIVA focuses on visualizing information from different perspectives. We can distinguish between two distinct phases of activities within this approach.

The first phase is to define and experiment with the perspectives. This activity is done mostly in solitude, although multiple users can share a primary model or a derived model. The purpose is to determine the information need and the relevant data for that need.

The second phase is that of multiple users collaborating by reviewing and discussing the different defined perspectives. This article focuses on the latter, the collaboration phase.

When a group of people collaborates, the group members must share a common workspace (Ellis et al. 1991). The task of a group of users is to interact with each other and present different views on shared information. Let us assume that the goal is to reach a decision, for instance, concerning which model to choose for a business process re-design project, as in the example of Section 2.

Sessions Collaboration normally takes place in some kind of meeting, which can differ in interaction protocols, group size, formality, etc. Each participant of the collaboration can have one or more roles depending on the sort of meeting. A role is a set of rights and obligations (Ellis et al. 1991). We distinguish the following roles: chair, listener, talker and interactor. The chair sets up the session, a listener is a passive par-
participant, a talker is actively explaining his arguments and, finally, an interactor is able to interact with shared resources. The rights and obligations of the different roles are determined by the interaction protocol. The possibility to switch roles dynamically is important, since a listener can change into a talker from one moment to the other.

Collaborative visualization in DIVA is a virtual meeting, where the participants are at different places and their desktops are connected by a network. We will call the event of such a virtual meeting a session. Session management should support several kinds of sessions and thus be able to handle changing numbers of participants, their roles and interaction protocols.

The notion of subgroups makes it feasible to split the total group of participants in (non disjoint) groups. These subgroups can communicate separately or perform subtasks. Subgroups can come into existence dynamically.

Sharing perspectives It is important that the cooperators can show their personal perspective or view to other participants, in order to support their arguments in a discussion. One way to share views is for one participant to enforce his perspective onto another user or group. Views can also be shared by means of a perspective repository, where participants can select a perspective they would like to consider. The perspectives they can choose from, must be deposited by other participants. This implies that not every participant should have to create her own perspective before joining a session. Obviously, there is a need to maintain meta-information, explaining what the perspectives are about.

Interference versus non-interference The common basis for the collaborators is the primary model, for instance, embodied by a simulation. Several derived models can depend on the primary model, and derived model could be related to a number of presentation models. When collaborating, the common basis should be the same for all the cooperators at every moment in time. To assure consistency, it is best to have the simulation act autonomously without the slightest interference. We will refer to this as non-interference. Non-interference does not restrict the possibility for each user to create his own perspective in any way. It does prevent somebody from rewinding, changing parameters or restarting the simulation while others do not want or expect this.

However, the need to stop, rewind or change parameters in a simulation is imminent. Considering multiple what-if situations, for example, is necessary when looking at different re-design alternatives, each with its own set of parameters. One way to meet this requirement, while upholding non-interference, is to store all past events in a database or to allow copies with different parameters of the simulation to be started.

While interaction with the primary model should be avoided as much as possible, derived models can be used in a more flexible manner. Several derived models can be created, all depending on one primary model. While the primary model can be considered a common basis that should not be interfered with, the derived model can be seen as a common workspace that permits interference. All participants of
a session could use the same derived model or multiple derived models could be created, depending on the need to share information concepts or to be independent of the other users.

**Communications** Some form of user-to-user communication is necessary to enable collaboration. These communications can range from a simple chat tool or whiteboard to sophisticated audio/video conferencing tools.

Tools of interest for use with DIVA include telepointers, to point out things of interest, raising hands, to indicate someone wants to speak, and voting tools, to support decision making (Ellis et al. 1991).

### 4.2 Requirements

Taking into account the issues for collaborative visualization mentioned in the previous section, we can summarize the following requirements.

- Session management is needed to control the virtual meetings, including the participants, their roles and interaction protocols.
- The participants must be able to share their perspectives by enforcement and via perspective repositories.
- The primary model should be interfered with as little as possible.
- Additional communication support is necessary, but falls outside the scope of this paper.

### 5 COLLABORATIVE MULTI-USER VISUALIZATION ARCHITECTURE

The DIVA architecture is intended as a framework and should be flexible enough to incorporate new components. The components used in DIVA are generic software components which interact with each other. The architecture is distributed, meaning that its components can reside on different hosts on the network. Related collaborative architectures can be found in Bentley, Rodden, Sawyer & Sommerville (1994) and Reinhard, Schweizer & Völksen (1994).

#### 5.1 Software components

Figure 2 shows the main components of the architecture. In the following list, the three components that were already present in an earlier version of the DIVA architecture (Schönhage & Eliëns 1997, Schönhage & Eliëns 1998), are listed first. The last three components in the list extend DIVA.
The main components of DIVA are:

- **Generator** — embodies the primary model
- **Shared concept space** — information store that contains the derived model
- **Presentation component** — the actual information visualization
- **DIVA services directory** — central registering of components
- **Collaborative session manager** — overall coordination of virtual meetings
- **Local collaboration component** — local collaboration support

In a normal situation, one DIVA services directory and more than one of each of the other components could exist. A presentation component and a local collaboration component are present at the desktop of each participant. A short description is given for each of the components.

The **generator** embodies the primary model and generates all data needed for the visualization. Examples of generators are simulations of business processes. The generated raw data is transferred to the shared concept space.

The **shared concept space** stores information in an expressive and adaptive way. The information is contained in the form of concepts that are stored in a hierarchical manner. Each concept has one or more data properties that represent the information. The data properties are updated with data coming from the generator. The received data can be stored directly or can be used to compute and store derived information.

The **presentation component** actually visualizes concepts from the shared concept space. It makes use of gadgets, which are generic visualization primitives that present certain types of information. As an example, cone trees are primitives (gadgets) to visualize hierarchical information (Robertson, Card & Mackinlay 1993).

Examples of gadgets are a rotating object which indicates a certain action or a histogram which displays data.

The **DIVA services directory (DSD)** is the central directory component. DIVA components (or services) can register here, identifying themselves and giving their location. Once they are registered, the DSD can inform other objects about the availability and whereabouts of these services.
The collaboration session manager (CSM) coordinates components. It deals with interaction protocols, which means it knows about the participants and their roles, sharing perspectives, user to user communication and consistency.

The local collaboration component is directly connected to the session manager. It is present at each participants desktop and handles interactions and information related to a collaborative session. It may for example display a list of participants and offer communication facilities.

5.2 User Environment

Figure 3 shows a typical user environment. Outside of the user environment, two components are shown. A generator, which is a simulation in this figure, feeds data into the shared concept space. This is depicted by the fat arrow. These two components can be situated anywhere on the network. Most of the components in a user environment use the display. The local collaboration component displays information about the collaborative session. The presentation component displays the visualization and the controller displays its user interface.

Display agents From the shared concept space, information is being sent to display agents. These agents are present at the user environment and each of them maintains one gadget. The information they receive is transformed into a visualization. As an example, consider the visualization in Figure 4 on page 12. The line of three
puppets in front of the desk is a visualization gadget depicting a queue. When the display agent senses that the length of the queue increases by one, it accordingly places a fourth puppet on the screen.

The display agents also reside in the perspective repository. When a user requests a certain view, the agents that represent that view are cloned and moved to the user environment to build the perspective in the VRML world. Enforcing a perspective onto another user is accomplished by cloning and moving the display agents from one user to another.

Now why can we call these entities agents? There are a lot of definitions of agents (Franklin & Graesser 1996), and the question can be raised why a display agent is an agent. In other words, how does it differ from a standard program or software component? For one, the agents are autonomous, which means they execute on their own. Second, they react on certain input and subsequently act on their environment, namely, they sense information from the shared concept space and act on the VRML world. They have some goals they need to accomplish. Third, the display agents can communicate with each other, for instance, to discuss how to place the gadgets each of them represents on the screen (this is a future feature). Fourth, they have a domain which they have knowledge of. This domain is the visualization of information. Fifth, they act on behalf of a user. Users can give their preferences to a display agent, and the agent will take care of it. All in all, the display agent fits quite a number of definitions of autonomous agents given by Franklin & Graesser (1996).

Controllers

Every DIVA component can have a separate mobile controller. It can be moved from one environment to another, so it can be shared by several participants. The ability to use a controller depends on the role of the user. Participants can request a controller or the chair could appoint it to one of them.

Controllers can have several functions. For example, a simulation can be controlled by starting and stopping the simulation and changing certain parameters. A controller for a shared concept space can be used to create new computed concepts or to decide which data from the generator is selected.

5.3 CORBA and the Web

DIVA is designed as a distributed object oriented system. The DIVA components are written in C++ and Java, and can run on different platforms. We use the Common Object Request Broker Architecture (CORBA) to let our distributed objects communicate with each other. CORBA (Siegel 1996, Orfali & Harkey 1997) abstracts from hardware, operating systems and programming languages. By using the interface definition language (IDL) to describe the interfaces between components and by making use of the object request broker (ORB), distributed components are able to communicate.

Voyager (ObjectSpace 1997) is an agent ORB written purely in Java, which supports CORBA. Voyager allows us to use mobile objects, a feature which CORBA
does not have. We use Voyager to construct the mobile controller components. These components are able to "dock" at a user environment and can subsequently show their user interface on the screen to let the user interact with it.

We use VRML (ISO 1997) as the main visualization tool. The users are able to navigate through the VRML worlds by using a VRML-browser. The External Authoring Interface (EAI) makes it possible to control the VRML worlds dynamically via the Java and Javascript languages.

The visualization gadgets in the presentation component are represented by mobile display agents. These agents are constructed using Voyager. Display agents can also "dock" in a user environment and, in addition, get access to the local VRML world. They collect the needed information from shared concept spaces to build and maintain the 3D visualization.

The combination of CORBA and the Web enables access to information resources by means of HTML, Java and VRML (see also Rohrer & Swing 1997). For example, the simulation and shared concept space can be hosted on a Unix server while the presentation components are executed in a Web-browser on Windows client machines.

6 APPLICATION

Figure 4 presents a screenshot of the desktop of a decision maker participating in a collaborative session as described in the example of Section 2. We describe a scenario of how the user gets to this display.

The decision maker starts a Java and VRML enabled Web-browser and follows a link pointing to a DlVA server. The resulting HTML file will setup the user environment. First, the user has to log in, making available her name and network address, and after that she can choose from one or more sessions to join. Once the user enters a session, she will be assigned a role and then gets a default or enforced perspective. The VRML world showing her view is embedded in the Web page. Two Java applets will contain the session interface and the collaboration interface (these are not shown in the screenshot).

With a push on a button she is able to request a remote control, which will arrive at her environment (assuming that she is allowed to do so). Once the remote control arrives in the form of a mobile object, this object pops up a remote control user interface on the display. The user is then able to control the simulation that is associated with the remote control. In Figure 4 the remote control can be used to run, stop and reset the simulation. In addition, the speed of the simulation as well as three parameters can be changed. Changes to the simulation will accordingly appear in the visualization in the browser window.
7 CONCLUSIONS

This paper is based on our belief that (interactive) visualizations are useful arguments in decision making because they provide such quick insight into information by using the human perceptual abilities. By means of collaborative visualization, decision makers are able to discuss a shared information source, such as a business process simulation, to convince other users of their point of view.

Based on a discussion of issues in collaborative visualization, we have concluded that the following requirements are needed for our visualization architecture.

First, different perspectives are necessary because multiple users, with different information needs, require different views on the data. These perspectives can be created by means of shared concept spaces and presentation components that make use of display agents.

Consequently, an important requirement is the exchange of visualization perspectives, for example, by enforcement or by a repository of perspectives. The exchange
is achieved by cloning and transporting display agents, which in turn define how and what available information is presented to the users.

To manage the cooperative sessions, we have defined two collaboration components that handle the rights and obligations belonging to the roles of the participants.

As a final requirement, we have stated that interference should be avoided as much as possible because other participants are involved in the actions taken. On the other hand, interaction with a running simulation to evaluate some what-if situations is very powerful. Therefore, we have created remote control components to interact with simulations and other data generators.

In further research, we are investigating an extension of the shared concept space to store sessions that can be replayed at a later time. Additionally, we are planning a case study to determine for useful visualization primitives to represent business information.

REFERENCES


URL: www.msci.memphis.edu/~franklin/AgentProg.html


URL: www.objectspace.com/voyager/documentation.html


USA', ACM Press.

8 BIOGRAPHY

Bastiaan Schönhage is PhD-student at the Vrije Universiteit in Amsterdam and ASZ Research & Development. His research comprises the design and exploration of a flexible software architecture for dynamic information visualization on the Web (DIVA). His research interests include information visualization, and distributed OO software architectures.

Peter Paul Bakker completed his master thesis on the collaborative aspects of the DIVA-project. He is a pre-graduate student in Software Engineering at the Vrije Universiteit and he is doing an internship at ASZ R&D. His research interests include collaboration, VRML and agent-based ORBs.

Anton Eliëns is lecturer at the Software Engineering section of the Computer Science Department of the Vrije Universiteit, Amsterdam. His research interests include object orientation, hypermedia and distributed logic programming.
Part Three

Applications and Empirical Studies
Developing a Multimedia Product for the World Wide Web

Linda Lisle, Scott Isensee, and Jianming Dong
IBM Corp.
11400 Burnet Road
Austin, TX 78758

1. INTRODUCTION

Our group is responsible for developing the IBM HCI and Ease of Use web sites. The goals in developing these sites have included investigating and demonstrating appropriate uses of multimedia in web design. Through our experience developing these sites, research we have conducted, user feedback we have received, literature reviews, interviews with web site developers, and related activities, we have learned a number of valuable lessons. This paper summarizes the most important of those lessons.

2. SYNCHRONIZATION OF MEDIA

Synchronization of media is particularly difficult on the Web. Issues include speed of the processor on the client, speed of the internet connection, and load on the server. These unknowns make it difficult to select media attributes such as frame rates for video or animation and sampling rates for audio quality. Multimedia is not yet a standard element of Web browsers. It typically requires a plug-in.

We decided to provide our data in multiple formats. We used animation technologies such as Flash or Shockwave which could be used by those with fast connections and provided downloadable versions such as Java applications for users with slower connections. The illustration below shows an example of several frames from a Flash demo which gives a preview of an application which can be downloaded.

Figure 1. Animation
3. DO’S AND DON’TS

Through experience, testing, and user feedback, we have learned a great deal about what works well and what doesn’t in user interface design for the web. To institutionalize this knowledge for ourselves and others, we created a set of Web Guidelines which are available at http://www.ibm.com/easy/

Figure 2. Web guidelines

4. PROJECT MANAGEMENT

The difficulty of developing for the Web is increasing as the available media and technologies increase. Multimedia development requires people with greater ranges of skills: visual design, audio design, programming, etc. The activities of these people must be coordinated. Classical project management techniques such as Gant charts have proven very helpful. The Web provides a great communication vehicle for the project members. We coordinate our team through project management charts on our intranet site.

Figure 3. Gant chart

5. NAVIGATION
The web provides a hyperlinked information structure which users must navigate through. In our testing we found users would become "lost in hyperspace" jumping from link to link. We were able to change this experience from one of going from place to place to one where information is brought to the user in a single place. We accomplished this by using frames to reserve parts of the page for navigation and refreshing only the content frame as the user navigated throughout the site. The figure shows an example of a page layout which provides navigation panes that remain constant while a content pane is refreshed.

6. RISING LEVEL OF EXPECTATION

Users are coming to expect increasingly higher levels of quality in the media they encounter on the web. If the quality of presentation is not sufficient, you can’t capture the readers interest long enough to deliver your message. We have addressed this by employing professional visual designers and by developing leading edge visual and audio elements in cooperation with our research division.
A feasibility study for a mobile multimedia tourist guide

Helmut Eirund
Hochschule Harz
Friedrichstr. 57-59, D-38855 Wernigerode,
Tel: +[49]-[0]3943-659-200, eirund@fh-harz.de

Gerlinde Schreiber
TU Clausthal
Erzstr. 1, D-38678 Clausthal-Zellerfeld, Tel: +[49]-[0]5323-72-2402, gerlinde.schreiber@informatik.tu-clausthal.de

Abstract
This paper presents a mobile and interactive multimedia information system that enables the visitor to follow the tracks in the cultural landscape of the Harz. Its main issue is to guide the visitors of the Oberharzer Bergbaumuseum in Zellerfeld on a 3 km walk and to visualize the consequences of the 1000 year old mining activities in the Harz for the natural environment. The target group of our study is the typical Harz tourist, i.e. young families and small groups of hikers.

Our main objective is to introduce a multimedia system that is used and accepted by a wide range of non-uniform people, whose interest in the multiple ecological effects of mining on nature is raised effectively.

Keywords
Multimedia learning, tourist guide, wearable computer, feasibility study, evaluation
1 INTRODUCTION

The Harz is a mountainous region of central Germany with a rich cultural heritage based on its 1000 year old mining activities. The history of mining as well as the life and struggle of the miners and their families is exhibited in more than 20 mining museums, some of which provide access to original underground mining facilities. But the main evidence of the mining age can be found outside the museums in the natural environment: in the artificial lakes and trenches that help to control the water works; the large areas of fast growing pine trees, which have replaced the domestic beech; and the heavy metal specific flora which has settled on the unused waste heaps.

In a study with the mining museum association ("Verbund der Oberharzer Bergbaumuseen", Clausthal-Zellerfeld), funded by the german "Bundesstiftung Umwelt", we examined the following questions: How can this ecological perspective of the mining age be presented museum visitors? How can the perception of the visitors be made aware of the consequences of mining on nature?

We have developed an interactive, mobile multimedia system which guides visitors on a determined route outside the museum and informs them at nine information points of selected and specific topics concerning the relation of mining and environmental problems. The system supports the user in two ways: First it serves as a guide to the information points that are situated on the circle route. When the visitor arrives at an information point, the system augments the visible environment. With multimedia techniques the system overcomes the constraints of space and time to provide an expanded view of the information point. (More on the background of the project's application can be found in (Eirund, Marbach 98).)

This paper is structured as follows: chapter 2 introduces the requirements both of the visitors and of the museum. The questions discussed here are as follows: Do visitors want to go on a round trip with a wearable computer? What expectations do they have concerning the mode and technique of presenting knowledge? What are the requirements of the museum? In chapter 3, we give a short description of the concept and realization of the mobile multimedia system. The central statement of this paper is given in chapter 4. Based on our first feasibility study, we discuss: the way the visitors use the system, what characteristics of the multimedia application are best accepted by the different user groups, and how users learn with the system. In chapter 5 we summarize and discuss our future work.

2 REQUIREMENTS ANALYSIS

The main mining museum of the upper Harz ("Oberharzer Bergbaumuseum" in Clausthal-Zellerfeld) is interested in demonstrating the effects of the mining on nature - not by yet another indoor exposition, but via a walk to the effected places. The museum intends to encourage tourists to make their own discoveries by taking
a predetermined route. Visitors need information to help them recognize particular changes in the vegetation, artificial lakes and trenches, hidden entries to mines etc. This information should be given just at the time and place where the phenomena explained can be perceived.

Two constraints have to be handled: The first concerns extension of the interesting objects in the space, the other their extension in time. Some phenomena - like the development of the forest from a healthy mixed forest to a quickly growing monoculture of pines - can best be considered over a period of centuries. Others - such as underground ditches regulating the water level in the mines - are decayed too much to be opened to the public. Nevertheless they are an interesting document of the mining in the Harz whose effect on the water level in the lakes can still be perceived.

From these requirements of the museum we conclude: We need a mobile mmIS to accompany the visitors on a walk through the forest. There the visitors will be offered information relating to their actual position.

What about the visitors? Are they willing to walk through a natural setting wearing high-tech-equipment? A first poll yields the following answers.

60% of the visitors questioned (280 persons) are interested in a high-tech-walk under the following conditions:

The technical equipment should be "small and light" and should not hinder movement. It should not attract the attention of others. The visitors want a gentle introduction to technical requirements of the system, preferably in a personal way. They want to decide themselves whether to use the information offered. They refuse to continually concentrate on the computer and to strictly follow its suggestions (on extra walks for example). Instead they want to enjoy nature, merely equipped with some additional information on the surroundings. They want to control the speed of the tour and to choose the best time for breaks by themselves.

We develop the system following these requirements.

3 REALISATION OF THE MOBILE INTERACTIVE INFORMATION SYSTEM

To mediate the information on the focused topics (i.e. timber production, water power) in authentic places, the visitors are guided by our mobile system to predetermined information points, where they can execute the appropriate information module, directly referring to the actual position. The excursion is performed as a typical "walking-tour" with verbal and non-verbal assistance from the system.

The information modules are designed, following different scenarios known from computer based learning. Some use conventional multimedia presentations to demonstrate the change of the surrounding nature through the last centuries. Others provide virtual access to non-open places (e.g. beneath the respective information
point) that can be explored on the screen. At another information point we apply a kind of goal-directed learning: some evidences of old mines have to be detected on a small excursion through the forest. By combining these different scenarios we want to animate the visitors interest during a walk of at least one hour.

Mobility, interaction, tour guide and the access to the proper information at the correct information point are further, technical requirements to the hardware and user interface that will be described in the following.

**System conditions**
Many projects in the field of augmented reality use „headmounted displays“ and speech recognition systems to enhance and explain the real world impressions with computer produced artificial information (see e.g. (MIT 1998), (GAT 1998)). Although this technical approach is convenient, it does not seem adequate for the visitors (as stated in chapter 2). Therefore we use a more conventional hardware not attracting too much attention: a handheld booksize PC.

With the Stylistic 1200 (Fujitsu), available since late 1997, we employ a standard Win-PC with pen-interface. In this way, we are also able to realize the multimedia content with common tools (among others, we apply Macromedia Director) and to transfer data from the development platform in an easy manner. With its pen interface, a weight of less than 2 kg, a harsh environment case and an antireflective sunshade, the system is completely reliable outdoors (see figure 1). Audio speakers and a bright 16 bit color screen provide for a common experience even in small groups.

![Figure 1](image)

*Figure 1* With the mobile Pen-PC (Fujitsu 1200 ST) on the trail (Pen-PC, Pen, optional headphones and screen cover)
Localizing the proper information points

On his trip to the information points, the tourist is guided by "August Ey", a historical person from the last century from Clausthal-Zellerfeld, who is known to tourists from various other events.

During the walk, August Ey serves as a pathfinder, who offers two alternative modes: "follow the map" displays a map with the actual part of the route marked. In the mode "by view" the single steps of the trip to the next information point can be called up as photos with additional comments by August Ey. Figure 2 shows the UIF with August Ey and the interaction mechanisms.

As mentioned before, the visitor should search for information points in the landscape where he can access the respective information modules. With Global Positioning System (GPS) it is possible to locate the actual site with a precision of about 10-20 meters. This discrepancy can make the "synchronization" of the real view with the view given on the screen very problematic. The use of infrared signals that can be processed by the mobile system provides much more exactness.

This guidance method is used in several in-door projects (e.g. (ABTA, 1998),
But obviously the effort to install and maintain the sender unit and to provide an autonomous energy source is quite high.

In our approach, we apply a more pragmatic technique: Each single information point is marked with a special code (we use a sequence of five "1" or "0"). These codes, fixed on trees, are easily recognized by hikers as they resemble the well known signs for walking tours. The code must be found at the site (thus stimulating attention) and confirmed by touching the appropriate buttons of the UIF with the pen. Only the proper code (that is only given at this particular site) executes the information module. The screenshot of the code input is given in figure 3. It shows the lower right side of the UIF that is devoted to navigation.

**Figure 3** Input buttons for access codes after reaching the info-point (substitutes the marked guiding buttons)

**Structure of the information modules**

Each information module focuses on a special topic (e.g. water power, timber consumption). All modules are stand alone applications that can be easily changed, enhanced or reused (e.g. in a stationary point of information in the museum).

There are three parts that make up a module:

- First, August Ey and his "grand niece" (a contemporary person, only present via her voice) introduce the topic and its particular environmental problem in a dialogue.
- Then August Ey explains the content in more detail, supported by pictures, graphics, audio and animation. In this part, the user has the opportunity to ask independently for more information, skip or repeat parts of the module.
- In the third part, the "grand niece" comes up again and explains the ecological context according to the knowledge of our time.

August Ey and his grand niece represent the past and the present time. While August Ey explains old mining techniques and related issues the grandniece stresses
the current interest in ecological implications. Figure 4 gives the navigational and module structure of the system.

![Diagram of the navigational structure of the system](image)

**Figure 4** The navigational structure of the system

4 FEASIBILITY EVALUATION

For the feasibility study we have accompanied 30 visitors of the museum on the circular route.

*Users*

Who is interested in being one of the first users of (a prototype of) the mobile system and is willing to answer questions on his high-tech-walk? Not the original 60% of the visitors, who had indicated interest. Obviously it’s quite easy to announce one’s interest – but especially elderly people admit that they are afraid of the new technology and of using it in public. Young families and visitors who are acquainted with computers (either at work or at home) agreed to serve as test users. No one without computer knowledge participated in the study.
Handling

All visitors, who are interested in the system are capable to handle it. The test persons use the UIF and the pen properly. August Ey is accepted as a personal guide. Especially children are attracted by his moving face. August Ey’s grandniece is recognized as a present day person. Her dialogue with her deceased granduncle doesn’t irritate anybody (TV’s benefit!).

The interaction mechanisms „repeat“ and „next“ as well as the orientation with help of a map or of views don’t need further explanation. The „exit“-button seems to be too ominous to be ever pressed. The help function “?” is used in different ways. Some visitors test their understanding of the help even before leaving the museum, others start confidently their trip. This behaviour can’t be related to age /education of the visitors.

The code at the information points can easily be entered.

Using the system

On the walk to the first information point all users behave nearly the same. They test the interaction mechanisms and check the different navigation possibilities. At the information point the code is entered carefully and the respective information is followed in detail.

Afterwards the visitors proceed in different ways. One group - mostly people who consider themselves to be high-tech-amateurs - cling to the system. They select all the information offered because they fear to miss some important technical hint. The other group of testpersons uses the system according to their personal preferences.

Personal preferences generally concentrate an those phenomena that can be perceived in the surrounding nature. Different (verbal and nonverbal) media supporting each other are appreciated. Visitors recognize and welcome the changing learning scenarios. They listen to short and simple explanations and refuse to read more than headlines. Nevertheless the visitors are interested in more detailed information (they welcome hints on other exhibitions etc), but it seems inadequate when given on a walk.

Surprisingly the visitors like a selfmade video of an underground ditch much better than some professional presentations - they are not looking for entertainment but for personal experience.

Most interesting is the use of the system by families. While hiking is quite boring for children (the adults always read the map and decide on the way), hiking with a computer’s help seems to be exciting. The adults know the multimedia system as little as their children do – therefore the classical roles change and everyone enjoys an equal status. In general the children interact more freely with the system than their parents. Thus they discover quickly new information in the system and follow hints concerning surrounding nature. Sometimes the adults are infected and find themselves exploring the embankment of a medieval lake.
Groups of adults enjoy the system best when they pass the handheld computer to one another. Whenever the responsibilities change, people explore their new possibilities as well with the technique as in the surrounding nature.

A main objective of the project is to raise the users interest in the presented topics effectively. Point-of-Information terminals (e.g. in museum environments) attract many visitors, but reach only some minutes of interaction (Compania Media, 1998). With the mobile system users are strongly bind to the application for more than one hour: on the one hand they apply the pathfinder functionality as a very practical guide to find the proper trail and on the other hand the different learning scenarios (e.g. goal based modules, augmented reality information) keep the interest at each information point.

Comments by the museum's mining experts

Though the museum has initiated the mobile multimedia system, interviewed employees have some difficulties in appreciating it. They are used to tours guided by experts through the museum where the visitors just follow instructions. Those parts of the exhibition that can be explored by visitors on their own are mostly used while waiting for a guide.

Most of these employees criticize the quality of the explanations given by the system. The system uses very few technical terms. Instead it illustrates interesting phenomena through simple verbal and nonverbal phrases and encourages visitors to make their own discoveries. (For example we offer a picture of the old pavement of paths leading to mines. Nearby we cross such an old path. Visitors who decide to follow it will find the hidden entry of a mine.) These aspects of the system make it appear much like an adventure game - which does not fit the traditional offerings of the museum. Besides, the system attracts a different kind of visitor (especially children), who up to now don't get the attention of most guides.

The expert's criticism confirms the results of (Najjar 1996) on multimedia learning, where it is stated that multimedia learning seems more appealing and effective for people with low prior knowledge.

5 SUMMARY AND FUTURE RESEARCH

In this paper we presented a mobile, multimedia computer system that was employed as a personal tourist guide. The application focuses on the presentation of the changes in the natural environment as a consequence of the 1000 year old mining history in the Harz. As most of the visitors are novices to this topic, multimedia systems are especially suitable for this task. The approach of the Oberharzer Bergbaumuseums in Clausthal-Zellerfeld is to guide the visitors with a mobile system on a predefined trip outside the museum and "augment" the reality at the places of interest.
During our studies with small groups of visitors we have noticed interesting "intra-group communication". During the prototyping of the system the museum is equipped with only one handheld PC. Therefore we could not study the systems used by several groups in parallel (e.g. by a class), but only by single groups of at most 4 persons. We wonder whether there will be any special effects of "inter-group communication".

After finishing the prototype that will be enhanced by the feedback we have gathered, a final analysis on the feasibility of the system will be performed in the museum. Thereafter, we plan to propagate this approach to other Harz museums within an EXPO 2000 project (within the project "EXPO on the rocks").

During 1998 we will install an additional stationary system within the museum that should provide more detailed information with more interactive parts. An interesting question is, how both systems will concur: does an informal contact with the stationary system make less likely the refusal of a tour with the mobile system? Does a guided tour increase the tendency to interact more deeply on various topics in the museum, e.g. at the stationary system?

6 REFERENCES


7 BIOGRAPHY

Dr. Gerlinde Schreiber: 1985 diplom in computer science; until 1988 at Siemens AG, working on communication protocols and as a delegate to DIN and ISO; until 1994 research assistant at University of Oldenburg and doctoral degree on distributed systems; up to now post-doctoral fellow at University of Clausthal; research topics: computers and education.

Prof. Dr. Helmut Eirund: 1985 diplom in computer science; until 1988 at TA/Olivetti research lab, working on multimedia document management; until 1991 research assistant at University of Oldenburg and doctoral degree on multimedia document archival; until 1994 at OFFIS research institute, working on multimedia development tools; since 1994 professor for multimedia at Hochschule Harz, University of Applied Studies and Research.
Design and empirical testing of a checklist for the evaluation of multimedia software for children

Duda, S.
Humboldt-Universität zu Berlin, Institut für Psychologie
Ingenieurpsychologie
Oranienburgerstr. 18, 10178 Berlin, Germany
Phone +49 (0)30 285165-1
sabrina.duda@rz.hu-berlin.de, http://www.kids-soft.ipfb.de

Abstract
An expert checklist for evaluating children's software has been designed and validated on ten children in the second grade. In the first part of the investigation ten psychology students evaluated three edutainment games, all of different quality, by means of the checklist. Based on these results the items on the checklist were then analyzed and filtered (analysis of variance, item analysis). In this way the length of the checklist was reduced considerably. The remaining checklist items became the basis for calculating a new software evaluation score.

In the second part of the investigation seven and eight-year-old children played with the same edutainment games as the students. While doing so they were observed and subsequently interviewed by the author. A regression analysis was used in predicting the outcome of the observation of and interviews with the children using the newly calculated checklist results. The results show that it is possible to predict children's reactions to certain edutainment games by using the checklist.

Due to the low number of subjects and the use of only three different edutainment games, however, the results should be regarded as a kind of preliminary test indicating a general tendency.

Keywords
Software, multimedia, edutainment, children, evaluation, checklist, usability.
INTRODUCTION

More and more, children are recognized as a new group of software users. While software ergonomics dealt in the past mainly with software for adults, a new tendency is now arising whereby the special characteristics of the designing and usability testing of software for children are examined. Hanna, L., Risden, K., and Alexander, K. J. (1997) of Microsoft have developed guidelines for usability testing with children. Robertson, J. W. (1994) criticizes the lack of attention given to the usability of educational software. She proposes various methods of testing usability with children.

At the CHI-Conference in 1997 there was a special program for children called CHI-Kids (Druin, A., 1997). A Listserv list CHI-Kids exists as well.

For the past two to three years software architects have been attempting to produce educational software designed especially for children and boasting a high quality. The target group of the software industry is becoming younger and younger; there are even CD-ROMs available which are meant for three-year-olds to learn and play with. The new genre is edutainment software: it combines education with entertainment.

Earlier in the history of the industry many software firms blindly launched products onto the market with the hope that the catchword 'multimedia' would suffice to convince the consumer – a hope that was soon shattered; many firms disappeared from the market. Today, quality is becoming an increasingly important factor. 'Kindersoftware-Ratgeber 1998' (Feibel, T., 1997), a guide to children's software, surveys the vast market. German youth welfare departments publish brochures in which certain computer games for children are recommended. The 'Unterhaltungssoftware-Bewertungskommission' (an organization which oversees entertainment software, hereon referred to as USK) examines computer games and confers to them a rating comparable to the age restrictions imposed by the 'Freiwillige Selbstkontrolle der Filmwirtschaft' (or FSK, which oversees cinematic entertainment) in the motion picture industry. The USK focuses mainly on the problem of violence and pornography in software. The USK rating is commonly accepted. However, it is not to be understood as a recommendation in relation to quality. This fact is often overlooked by parents.

There are institutions and persons responsible for establishing the criteria by which computer games are to be recommended, such as the 'Arbeitsgemeinschaft Kinder- und Jugendschutz', an organization involved in the protection of children and young people (Lerchenmüller-Hilse, H., 1995), or Geisler, T. (1995) or Zey, R. (1994). The most extensive criteria were developed by Fritz, J. and Fehr, W. (1996) of the 'Computerprojekt Köln'. They are used for 'Computerspiele auf dem Prüfstand' (an examination of computer games), a series of booklets published by the 'Bundeszentrale für politische Bildung' (a German federal bureau for political education). These criteria place emphasis on the contents of the games and on the pedagogical aspect pertaining to them.

The 'Institut für Medien und Bildung' (Institute for Media and Education) confers a seal of approval to multimedia software of high didactical value. The criteria by
which this value is measured focus on the educational aims and user-friendliness (source: L.A. Multimedia - Magazin für Medien und Bildung, 1997; no author mentioned).

At present there is no criteria catalogue available to provide a detailed rating of the quality of various edutainment software. But there is a great need for such catalogues and official software ratings. Some software firms are interested as well in having the quality of their products tested.

2 ISSUE

In order to address the need for an evaluation instrument, the purpose of this work has been to develop a checklist for evaluating children's software. An effort was made to develop a general checklist which can be applied to various types of software. This study deals with edutainment software; that is, software with which children aged four to ten can play and learn. The checklist can be adapted so as to be suitable for purely educational software.

The checklist is designed to be able to differentiate between three edutainment games of different quality. In addition, it is to be validated by means of empirical testing with children. The checklist scores are to correlate with the children's behavior and their interview scores. Games with a high score on the checklist should therefore be preferred by the children.

3 CHECKLIST DESIGN

A test version of the checklist, with 236 items, was developed.

The aspects peculiar to evaluating software which is neither designed for office work nor meant to serve as a tool for carrying out assignments had to be considered. Edutainment software is a 'tool' for having fun and learning. To achieve both the user has to in fact carry out assignments given in the software, such as finding hidden things, solving problems, or training his sensorimotor abilities; but not in the sense of being an employee. The dimensions of conventional software evaluation as described for instance in EVADIS II (Oppermann, R.; Murchner, B.; Reiterer, H.; Koch, M.; 1992) or the ISONORM 9241/10 checklist (Prümper, J. & Anft, M., 1993) were not suitable for evaluating children's software and so a new instrument had to be constructed.

A second source of complication was the fact that the users in this case are children. It is not possible to ask a child a large number of detailed questions about such aspects of software as usability. Therefore, adult experts have to evaluate this kind of software and the problem of perspective arises. It had been decided that the adults would judge the software from their own personal point of view. Some (very few) questions require that the adults answer from the perspective of a child. When these questions occur this is explicitly mentioned.
Dimensions of the checklist

- **Cover and booklet**
  Does the cover contain all important information? Is the booklet (a very thin user’s manual with information for the parents) well written?

- **Entertainment value**
  In software for children, especially edutainment software, the entertainment value plays a great role. If the software does not entertain, children will not engage themselves in it, and no educational effect will be achieved.

- **Suitability for children**
  The design of the software should take into account the special needs of children. Tasks, for example, should be suitable for children, and the child should be able to identify with the characters.

- **Ease of use**
  Usability is a central factor in children’s software. Without a clear, consistent design, appropriate feedback, and a help function, the joy of playing is spoiled.

- **Load**
  Children’s software should not of course cause stress by pressuring the users to accomplish a task within a very short period of time, or by exhausting their sensorimotor skills.

- **Educational value**
  The educational value is more or less explicit in children’s software. There is software which conveys facts or information usually acquired at school and there is software which implicitly encourages general problem-solving. (And of course all software products serve as a means of learning how to use the computer.)

The items have a rating spectrum of five degrees. Some items (e.g. Does the game have a tutorial?) can only be answered with yes or no.

4 EXPERIMENT I – CHECKLIST (STUDENTS)

4.1 Method

**Subjects**
Ten undergraduate psychology students - five women and five men. The average age was 21.8 years.

**Material**
The three edutainment games selected for the study were:
- Max und das Schloßgespenst (Tivola).
- Gus geht nach Cyberopolis (TI New Media).
Procedure and design
Each student judged all three edutainment games on the basis of the checklist. The order in which this was done was balanced. The subjects played each game for approximately one hour. Afterwards they answered the questions on the checklist and returned to the game when necessary. The testing of one edutainment game took approximately three hours.

4.2 Analysis and results
The items were coded from 1 to 5; positive = 1, negative = 5. (The 'yes/no' questions were coded 1 and 5 respectively.)

Item selection
The item selection was carried out in two phases:

1. ANOVA (analysis of variance)
   For each individual item the play factor and the subject factor were checked in terms of exerting a significant impact. Ideally, the play factor has a significant influence and the subjects do not. The items for which the play factor exerted a significant influence (while the subject factor did not) remained on the checklist.
   After this analysis 89 items were filtered from the original 236 items.
2. Item analysis
   Items which were not selective, i.e. items with an item discrimination index of less than 0.3, were eliminated.
   Next, the difficulty of the items was analyzed. Only items having a difficulty index of between 0.2 and 0.8 were accepted.
   55 items remained.

The new checklist
The original checklist was reduced from 236 to 55 items.

The following is a description of the remaining items (the questions were originally posed in German):

• Cover and booklet
  Cover information.
  Booklet: clarity, thoroughness, presence of examples, explanation of educational aims, motivation value.

• Entertainment value
  Whether the following are offered: fun, things of interest, high number of animations, varying animations, atmosphere, background story, objects of game, long playing time, varying feedback, sufficient praise, performance checks, chance to succeed via player’s achievement, quick screen display, attuning of music and sound to game actions; and what the general entertainment value is.
• **Suitability for children**
  Whether the game appeals to a child, how appropriate the tasks are for a child, how appropriate the text is for a child.

• **Ease of use**
  Whether the following are present: highlighting of icons when 'touched' or clicked on, visibility of text against background, overview of game rules, explanation of game rules, method(s) of navigation, sufficient online help, help function, help function in every situation, does the game facilitate remembering the important things, adaptability to personal needs, different levels of difficulty, method of stopping or skipping processes within the game; and what the general ease of use is.

• **Load**
  Whether the demands of the game can be fulfilled, how heavy the cognitive load is, and whether there is a balance of demands placed by the game.

• **Educational value**
  Whether the following are promoted: different abilities, independent thinking, the child’s reading his or her first individual letters or words, the learning of foreign words or sentences, communication abilities, cooperation.
  Whether the tasks are: various, meaningful, of educational value.
  Whether text is: read aloud, read in a way that a child can follow the text on the screen.
  How high the educational value in all is.

**Checklist results**
The checklist results were based on the 55 items selected. The arithmetic means of all the items were computed. The items were not yet weighted.

The arithmetic means (low values are positive, high values are negative) were:
- **Max** 2.36
- **Gus** 2.04
- **Zuppel** 3.57

'Gus' attained the best score, 'Max' the second best, and 'Zuppel' the worst. All differences between the games were significant.

**Interrater reliability**
The interrater reliability was 0.92 (one subject had a correlation of only 0.66 with the nine other subjects; without this subject the interrater reliability would be 0.98).

**4.3 Discussion**

Only those items which prompted different results for the three games remained in the final checklist version. The difficulty here is that the three selected games did not differ in all items and all dimensions. Many of the questions in which the games did not differ are very important, e.g. the question regarding violence content, as well as most of the questions in the dimension 'load'. None of the three
games exposes the user to violence or is excessively demanding. Thus the items representing these aspects were eliminated.

Owing to the importance of some of the 236 original items they should not necessarily be ignored. Items having a low selectivity or which reveal extremely high difficulty scores can also prove useful for the evaluation. With as few as three games being employed in the study, it cannot be expected that all the items succeed in defining each game separately and that all five possible answers for each item be checked with equal frequency.

The results obtained from the checklist, now containing 55 items, show significant differences between all three games. The game 'Gus' was most preferred by the adults. This is probably due to the fact that it offers the highest educational potential.

These results must be conservatively assessed because of the low number of games tested and owing to the fact that the results were influenced by the selection of the three games.

Figure 1  Checklist results (high values indicate insufficient quality, low values indicate high quality).

5 EXPERIMENT II – BEHAVIORAL DATA AND INTERVIEW (CHILDREN)

5.1 Method

Subjects
Ten pupils of the second grade took part in the study. They were between seven and eight years of age and comprised nine boys and one girl.
Material
Max und das Schloßgespenst (Tivola).
Gus geht nach Cyberopolis (T1 New Media).

Observation program
To simplify the study, a VisualBasic 3.0 program was written for the observation of the children. On the program surface 19 buttons with the observation variables could be seen. Whenever a subject behaved in a manner corresponding to an observation variable, the appropriate button was then clicked with the mouse. The program counted the clicking frequency and computed the quotient of positive and negative observations.

- Positive observations
  Laughter, smile, joy, exclamations/sounds (neutral), comment to observer, comment to computer, comment to oneself, pride, fascination.
  Commenting shows that the child is interested in the game and is therefore to be judged positively.

- Negative observations
  Aggravation, anger/rage, disappointment, sigh, disquietude, question, helplessness, looking away, boredom, exhaustion.

- Additional buttons
  A pause button (for every kind of pause, also implemented when subject wanted to take a break).
  A button for entering text.

Interview
When the game was over the child was asked fourteen questions in connection with the following factors: fun, identification, subjective achievement, difficulty, exhaustion.

Procedure and design
The children were called in from their school lessons one by one. In a room at the school they played 'Max' and 'Gus' on a multimedia PC for about 50-60 minutes and 'Zuppel' for about 20-30 minutes. (The game 'Zuppel' requires only 20-30 minutes to play.) While playing the children were observed by the author with the help of the program. After each game they were interviewed for 15 minutes. The sequence in which the games were played, as well as the time of day (a.m. and p.m.), was balanced. When all three games were over every child was asked which game he or she liked most, second most, and least (preference judgement).
5.2 Analysis and results

Preference judgments of the children

- Seven children preferred 'Max'.
- Two children preferred 'Gus'.
- One child could not decide between 'Max' and 'Gus'.
- Eight children liked 'Zuppel' least.

Interview

The interview questions were credited with 0, 1, or 2 points. Arithmetic means were computed for every game.

Only the differences between 'Max' / 'Zuppel' and 'Gus' / 'Zuppel' were significant.

<table>
<thead>
<tr>
<th></th>
<th>Max</th>
<th>Gus</th>
<th>Zuppel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1.74</td>
<td>1.76</td>
<td>1.52</td>
</tr>
</tbody>
</table>

Behavioral data

The program computed the frequencies. The variable 'look away' was not retained because the children looked away from the screen to contact the observer.

The quotient was computed from the remaining positive and negative observations. All children displayed more positive than negative behavior. The one extreme value (22.67: almost four standard deviations from the average) was excluded. Thus data from nine children were used. The behavioral data appeared to correlate with the children's preferences since 'Max' had the highest value. But statistically only the differences between 'Max' / 'Zuppel' and 'Gus' / 'Zuppel' were significant.

<table>
<thead>
<tr>
<th></th>
<th>Max</th>
<th>Gus</th>
<th>Zuppel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>6.08</td>
<td>4.48</td>
<td>2.81</td>
</tr>
</tbody>
</table>

5.3 Discussion

The results of the interview and the observation were rather similar. In both cases the only significant differences were between 'Max' / 'Zuppel' and 'Gus' / 'Zuppel'. 'Max' and 'Gus' scarcely differed statistically from each other.

The results of the observation were more clear than those of the interview. Observation seems to be a very suitable method when children are test subjects.

The preference question ('Which game did you like best?') produced the clearest results and demonstrated the preference for 'Max'. Unfortunately, these results could not be used for the statistical analysis.
6 CORRELATION BETWEEN CHECKLIST AND CHILDREN'S DATA: REGRESSION ANALYSIS

The regression analysis shows the type of correlation existing between two variables. Thus the value of the dependent variable – the behavioral data and the interview – can be predicted by using the value of the independent variable – the checklist.

If it is possible to predict the children’s data by using the checklist results, the checklist will be validated.

*Linear regression analysis*

There is scarcely a correlation between behavioral data and interview results. They apparently measure two different things. Therefore, two analyses of regression were computed. The behavioral data seem to represent the true feelings of the children much more so than the interview data. The differences between the games are more prevalent in the behavioral data. Observing behavior is a generally more suitable method for young children than interviewing.
6.1 Behavioral data

The checklist results of all ten students and the behavioral data of nine children were used for the regression analysis. The arithmetic mean for each game was computed to produce three checklist values and three behavioral values.

The following equation was computed:

Behavioral data = -1.54 * checklist results + 8.55.

\[ y = -1.54 \times x + 8.55. \]  

The checklist values are reversed the polarity of the behavioral data. A high checklist score is a negative value, a low checklist score a positive value. The straight line is therefore drawn from top left to bottom right and does not begin at the origin. The coefficient \( b \) in the equation therefore has a negative sign.

The \( r^2 \) shows the quality of the adaptation of the regression line. It represents the proportion between explained variance and total variance (sum of squares regression / sum of squares residual). \( r^2 \) is always between zero and one. In this case the adaptation was satisfying: \( r^2 = 0.58 \).

![Regression analysis](image)

**Figure 3** Regression analysis (behavioral data).

6.2 Interview

The checklist results of all ten students and the interview data of ten children were used for the regression analysis. The arithmetic mean for each game was computed to produce three checklist values and three behavioral values. In this case the adaptation was very good: \( r^2 = 0.98 \).

The following equation was computed:
The results of the regression analysis must be considered based on the fact that only three games could be examined and therefore the line consists of only three points. The two different regression analyses demonstrated different qualities of adaptation. The children’s interview data correspond most to the checklist results. The children’s behavioral data were less successful although the adaptation was satisfactory.

Perhaps answering questions – whether in an interview or with a checklist – produces similar data while observation of behavior basically offers a completely different quality of data. One could speculate that observing the behavior of adults may provide values similar to those obtained when children are observed.

7 TEST CRITERIA

The design of the checklist is based on the principles of the classic test theory (CTT) model.

Reliability (internal consistency)
Cronbach’s alpha = 0.97.

Validity (criterion-related validity)
Correlation coefficient = 0.76 (criterion = children’s behavior).
Correlation coefficient = 0.99 (criterion = children’s interview).
8 GENERAL DISCUSSION

The results allude to a few general problems involved in constructing a checklist for adults to evaluate children's software. The preferences of the adults and those of the children differed slightly. The adults preferred 'Gus'. Some of the aspects on which the adults put emphasis were different from those placed by the children. Thus, the high educational value of the game 'Gus' seemed to be the main appeal for the adults. The children clearly preferred 'Max' when being asked directly which game they liked best (eight of the ten children). This result was also apparent in the behavioral data although there it was of no statistical significance.

In the interview, no difference at all occurred between 'Max' and 'Gus'. In both children's data – the behavioral data and the interview – there were no significant differences between 'Max' and 'Gus', only between 'Max' / 'Zuppel' and 'Gus' / 'Zuppel'.

In order to standardize the checklist, ten to fifteen edutainment games and roughly twenty adults and children would be necessary. It would also be useful to have software experts rather than students for the study.

The study must be seen as a kind of exploratory study indicating a tendency. The results are connected with the selected games: the elimination of the items is related to the three games.

The regression line shows that it is generally possible to predict the children's values with the checklist. Thus an expert would be able to judge the quality of children's software with the help of the checklist. He or she could roughly predict how a child would react to a certain software. The checklist could also be used as a guideline when designing software for children.

However, the involvement of children in the designing and evaluating of children's software is indispensable for a well-balanced decision.

9 OUTLOOK

For a practical application the checklist procedure must again be considered. A three-step scale would perhaps be more appropriate. The items must be weighted. Furthermore, they could be summarized to different modules so that the checklist could be adapted to different kinds of software by selecting only certain modules.

A validation with a greater number of subjects and edutainment games would be desirable.

10 REFERENCES


11 BIOGRAPHY

Born in Munich in 1968. Study of psychology in Regensburg and Berlin. Advanced studies in engineering psychology and minor in computer science. Special interests: software for children, software ergonomics, the Internet, computer science.
Performance Evaluation of Input Devices in Virtual Environments

Andreas Roessler, Volker Grantz
Fraunhofer IAO
Nobelstrasse 12, 70569 Stuttgart, Germany,
Andreas.Roessler@iao.fhg.de

Abstract
The user interface approach of virtual reality promises to be superior to two-dimensional approaches. Therefore, there is a need to perform experiments with different input devices. We developed a virtual environment test bed which integrates different input devices and modules for rapid modelling tests and evaluation. Our focus of the tests was a comparison between a conventional computer mouse, a space mouse and an electromagnetically tracked device. With the tests, we tried to measure the accuracy and performance of grabbing and positioning virtual objects.

Keywords
usability, evaluation, virtual reality, input devices, 3d tracking

1 INTRODUCTION

Until today there are very few highly interactive applications of virtual reality technology. Most of the applications around the globe focus on the visualisation of information – the used interaction is restricted to simple walk-throughs. As it happens quite often even the design of the „simple“ walk-through-interaction is too difficult for normal users.

VR-evangelists even dream of highly interactive applications where the immersed user is able to interact with any kind virtual object very intuitively. To meet this goal, we need to understand the characteristics of virtual environments, three-dimensional input devices and the basic tasks that have to be performed.

Like other VR groups, who develop various interactive applications for industrial partners and for research purposes, we urgently need this kind of better understanding of man-machine-interactions in virtual environments.

In a first series we tested the ergonomic issues of VR systems /Deisinger and Riedel 1996/ and user interactions in a CAVE-like projection environment /Blach, Simon and Riedel 1997/. The aim of the tests described in this paper is the evaluation of different input devices and an analysis of their characteristics. Design of Experiments
1.1 Aspects of Evaluation

The objective of our experiments was to identify characteristics of different input devices. The task to be fulfilled were grabbing and accurate positioning of virtual objects. We were especially interested in the following aspects:

- Efficiency
  How fast can users move virtual objects to reference positions?
- Accuracy
  How accurate do the final positions match the objects with the references?
- Users’ Satisfaction
  What are the users subjective opinions concerning the input devices?

1.2 The Input Devices and their Interaction Modes

As evaluation devices for our test we selected the standard computer mouse, the spacemouse and a simple self-designed device, consisting of a sensor of an electromagnetic tracking system and a button. We did not include an dataglove because of the following reasons: First, the test does not focus on grabbing the objects but on their positioning. Second, there are ergonomic reasons which exclude the datagloves from broader use: all available datagloves have only one size – at least in our lab. Therefore, the gloves do not fit for very small and very big hands, and need to be calibrated by the software. Additionally, taking on and off the gloves is inconvenient for the users.

*The Computer Mouse*

We included the mouse (see figure 1a) in our test because there are a lot of users who are familiar with it as well as many CAD and modelling packages which use the mouse as a three-dimensional input device. The typical problem is how to map all six degrees of freedom to a two-dimensional device. For the test, we mapped the movements as shown in the following table:

<table>
<thead>
<tr>
<th>Pressed button(s)</th>
<th>Movements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Button</td>
<td>Forward &amp; backward, left &amp; right</td>
</tr>
<tr>
<td>Right Button</td>
<td>Up &amp; down, left &amp; right</td>
</tr>
<tr>
<td>Left &amp; Right Button</td>
<td>Heading + &amp; -, pitch + &amp; -</td>
</tr>
<tr>
<td>Middle</td>
<td>Grab an object, if collided</td>
</tr>
</tbody>
</table>

*The Spacemouse*

The spacemouse (see figure 1b) is a commercially available three-dimensional input device which is frequently used for CAD. With a spacemouse, the users are able to define the speed of movement and rotation of virtual objects by pressing a slightly movable half sphere. The half sphere can be pushed left, right, forward, backward, up and down, and the grabbed virtual objects move accordingly. By applying a torque to the half sphere, the virtual objects rotate. The speed of rotation and the position movements of the objects are set proportionally to the applied force and torque.
The spacemouse offers six programmable buttons; we used button 1 for grabbing virtual objects and button 3 to reset the orientation. Additionally, the spacemouse includes special features by hardware, e.g. an input mode which disables all degrees of freedom but the one with the highest input values. As the other tested input devices do not include that kind of assistance, we did not use these modes for our tests.

**The Tracked Button**

The tracked button (see figure 1c and 1d) is a simple device designed by the author, consisting of a sensor of the electromagnetic tracking system and a button. The button is polled by a micro-controller, which is connected via RS232 with the host computer. The purpose of the button is to grab virtual objects. The tracking system provides absolute values for position and orientation relative to a physical reference point which is set by the electromagnetic transmitter. A logical reference point was set in a comfortable initial position defined by the subject.

![Figure 1a the mouse](image)

![Figure 1b the spacemouse](image)

![Figure 1c the tracked button](image)

![Figure 1d a subject with the tracked button during the test](image)
1.3 Hardware & Software

The tests were developed with the VR-kernel Lightning /Blach et al 1998/ , a development of Fraunhofer IAO. It consists of a rendering engine based on SGI Performer, device drivers for input and output devices, a routing manager that controls the application and a C/C++ application programming interface. Additionally, it includes a high-level scripting language, which is an extension of TCL (Tool Command Language /Ousterhout/). The complete test suite including the definition of the input and output devices, the interaction modes and the protocol recording was realised in TCL. For the control of the test suite by the supervisor, a GUI was built with TK, another extension of TCL. The GUI and Lightning communicated via TCP/IP. As a hardware platform we used a SGI Onyx with two RealityEngines2 and six processors. The tracking device was a Motionstar Extended Range from Ascension.

1.4 The Test Environment & Realisation

The virtual environment (see figure 2) was presented to the user wearing shutterglasses on a large screen (3m x 2m) stereoscopic video projection (Barco, resolution 960x960, frequency 96 hz). The virtual viewpoint was not tracked but was fixed by the test supervisor. The subjects sat (with mouse or spacemouse) or stood (with the tracked button) in front of the screen.

Before the test, the subjects had to fill in a questionnaire, that helped us to check their specific experience. After the test, we asked them to name the preferred input device.

The test included four similar tasks: the subject had to grab one of the four objects from the foreground and to move it to the reference position in the background. The first object was a sphere where rotation was not considered. The other objects had to be positioned and oriented to cover the reference objects perfectly.
The subjects had to move the object with each of the considered input devices. Before the tests, the supervisor explained the devices and the users had the opportunity to get acquainted with the input devices and with the grabbing of the virtual objects. All subjects performed the tests in the same order: mouse, spacemouse and last the tracked button. 

The used interaction technique was the same in the three tests: the input device directly controlled a three-dimensional cursor, represented by virtual tongs. An object could be grabbed when the cursor collided with it. The collision and the successful grabbing was visualised by a change of colour and shape of the virtual tongs. 

In total, twenty subjects performed the tests. All of them had used the computer mouse for mainly two-dimensional applications before, none of them had experience with the spacemouse and three subjects were familiar with the use of a tracked button. 

All movements of the cursor and all grab actions were recorded in a protocol file for evaluation purposes. From the files, the resulting period and accuracy was calculated automatically. 

Similar tests had been performed by Hinckley et. al. /1997/, who compared pure rotation of virtual objects with mouse and tracked devices. Poupyrev et. al. /1997/ developed a framework for the evaluation of immersive direct manipulation, but did not focus on the comparison of devices.

2 RESULTS

The following results were received from an analysis of the protocol files recorded during our test sessions. We tried to obtain information on efficiency and accuracy from the files and subjective judgement of the users.

2.1 Measured Values

Figure 3 shows the average time the users needed to complete the tests. It shows, that the use of the tracked button is significantly faster than the two other devices.

![Figure 3](image-url) 

*Figure 3 average total time in seconds*
Figures 4 and 5 compare the positioning of object 1, which did not need rotation, and object 5. It is interesting, that the interaction with the tracked button took approximately the same time even when the rotation was needed. The time difference between tracker button compared with mouse and spacemouse results mainly from the difficulties in orientating the objects correctly. The results fit to Hinckley’s /1997/ who concluded that rotation of virtual objects with tracked devices is 36 % faster than with the mouse.

Table 2 average deviation and range for positioning of object 3 (cylinder), X-,Y-,Z-axes

<table>
<thead>
<tr>
<th></th>
<th>Mouse</th>
<th>Spacemouse</th>
<th>Tracked button</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average deviation X</td>
<td>0,249</td>
<td>0,394</td>
<td>0,42</td>
</tr>
<tr>
<td>Range of deviation X</td>
<td>0,112</td>
<td>0,089</td>
<td>0,165</td>
</tr>
<tr>
<td></td>
<td>0,386</td>
<td>0,699</td>
<td>0,675</td>
</tr>
<tr>
<td>Average deviation Y</td>
<td>0,997</td>
<td>0,76</td>
<td>1,028</td>
</tr>
<tr>
<td>Range of deviation Y</td>
<td>0,062</td>
<td>0,277</td>
<td>0,52</td>
</tr>
<tr>
<td></td>
<td>1,932</td>
<td>1,243</td>
<td>1,536</td>
</tr>
</tbody>
</table>
The results of the positioning (see table 2) show that sufficient accuracy (for the subjects) was reached in similar ranges by all input devices. Table 3 shows the deviation range for the orientation of two objects. The tracked button has the lowest absolute deviation and the lowest variation between different users.

Table 3 average deviation range for orientation of object 3 (cylinder) and 5 (cone)

<table>
<thead>
<tr>
<th></th>
<th>Object 3</th>
<th>Object 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heading</td>
<td>Pitch</td>
</tr>
<tr>
<td>Mouse</td>
<td>0,699</td>
<td>1,771</td>
</tr>
<tr>
<td></td>
<td>11,568</td>
<td>9,457</td>
</tr>
<tr>
<td>Spacemouse</td>
<td>1,119</td>
<td>1,465</td>
</tr>
<tr>
<td></td>
<td>7,729</td>
<td>6,587</td>
</tr>
<tr>
<td>Tracked button</td>
<td>1,194</td>
<td>0,588</td>
</tr>
<tr>
<td></td>
<td>5,93</td>
<td>5,584</td>
</tr>
</tbody>
</table>

2.2 Assessment of the Users

After the test all users filled in a questionnaire in order to assess their subjective opinion on the different input devices. 85 percent (17 users) preferred the tracked button, 10 percent (2) the spacemouse and one user (5%) preferred the standard mouse. The following table shows the answers of the subjects on specific usability criteria:

Table 4 Average subjective assessment of the users (values range 1 – 6, best: 1)

<table>
<thead>
<tr>
<th></th>
<th>Mouse</th>
<th>Spacemouse</th>
<th>Tracker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>2,632</td>
<td>2,842</td>
<td>3,052</td>
</tr>
<tr>
<td>Efficiency</td>
<td>4,263</td>
<td>3,421</td>
<td>1,368</td>
</tr>
<tr>
<td>Overall usability</td>
<td>3,579</td>
<td>3,158</td>
<td>1,737</td>
</tr>
<tr>
<td>Learnability</td>
<td>3,368</td>
<td>3,105</td>
<td>1,368</td>
</tr>
</tbody>
</table>

Again, the users ranked the tracked button as number one in the criteria efficiency, overall usability and learnability. They were not happy with the accuracy of the tracked button – an answer which does not match with the measured values.
3 CONCLUSIONS

We know from discussions with experienced spacemouse users, that this input device can be very efficient and accurate. Nevertheless our tests show, that new users definitely prefer the tracked button which seems to be much more intuitive and efficient. The reason is the direct mapping between the orientation of the tracked button and the virtual object. The assessment by the (inexperienced) subjects shows, that they dislike especially three-dimensional rotation with mouse and spacemouse in combination with the used mappings. A paradox conclusion is that the tracked button is both accurate, shown by the measurements, and inaccurate, perceived by the subjects. Therefore, our research focus will be the improvement of the accuracy and the ergonomics of the use of tracked devices. Additionally, we will consider an aspect that has not been included in our recent tests: In order to grab objects which are not within reach we need to implement and test additional features for the use of a tracked button.

4 LITERATURE