

Workshop
**»Hypersystem-
Konzepte in Medien
und kultureller
Produktion«**

15. bis 17. Juli 1991
Universität Lüneburg

Vortragspapiere

Heiko Idensen, Matthias Krohn: „Entwurf einer Hyperpoetik“

Peter Bøgh Andersen, Berit Holmqvist: „Vector spaces as the basic component of interactive systems. Towards a computer semiotics.“

Frank Voeltz, Thomas Tesch, Tom Quellenberg, Ulf Hönisch: „Ansätze eines hypermedialen Filmkatalogs“

Rita Abel, Jörg Blankemeyer, Wolfgang Coy, Jörg Geißler, Ferruh Mercan, Ingo R. Meier, Oliver Schormann, Antje Tietjen, Torsten Rahn: „HYDRA – Filmanalyse im rechnergestützten Medienverbund“

Peter Gendolla: „Literatur und Kunst - Vorbereitungen für eine Ästhetik der Hypersysteme bei R. Roussel, H. Bellmer und O. Wiener“

Kurd Alsleben (Interdisziplinäre Computerei der HBK Hamburg): „Dialogische Künstlerrolle (von HandlochKart' zu Hai'pe:ka:d)“

Fahri Yetim: „Integration von Expertensystem- und Hypertext-Techniken am Beispiel des Systems WISKREDAS“

Annette Meini: „HyperXPert – Hypertext und Expertensteme“

Paul Ferdinand Siegert, Walter Uka, Martin Warnke: „Das Projekt EbsKart – Medienmix in einem interaktiven Auskunftssystem“

Karl Clausberg: „Medienmix – Im Kopf und vor Augen“

Erich Röhrich, Jürgen Taeger: „Gewässerschutz - ein integriertes Lernprogramm und Informationssystem zum Umweltrecht“

Ingo Carl, Aly Sabri: „Anwendung von Hypersystemen in der Medizin“

Norbert Bolz: „Navigation im docuverse. Perspektiven einer hypermedialen Darstellung von Komplexität.“

Heiko Idensen
Pool-Processing
ENTWURF EINER HYPERPOETIK

Zusammenfassung:

Basierend auf den Erfahrungen des Projekts Pool-Processing, das durch verschiedene offene HyperMedia-Environments, Installationen auf Medienfestivals¹, Multimedia-Performances², Workshops, in Forschungsprojekten und nicht zuletzt auch wiederum mittels linearer Texte³ eine offensive ästhetische Entwendung informationsverarbeitender Medien betreibt, (siehe beiliegende Dokumentation) wird bezugnehmend auf literarische und audiovisuelle Experimente⁴ eine Poetik von hypermedialer Produktionsweisen entworfen.

Ausgehend von den Konzepten des offenen Kunstwerks, innerhalb dessen Autoren, Interpreten und Zuschauer in einem Netzwerk unerschöpflicher Beziehungen interagieren, dialogischer Literaturkonzepte, die eine produktive Rezeption provozieren, sowie intertextuellen Produktionsweisen von Texten, die auf produktionsästhetischer Seite kooperative/kollektive Textproduktion als Grundlage eines Textbegriffes im Sinne auch eines sozialen Netzwerks ermöglichen, werden dynamische Operationen hypermedialer Verknüpfung und Vernetzung von Texten/ Bildern/Aktionen/technischen Bildern/Programmen/Ideen als ästhetische Qualitäten einer HyperPoesie ausgemalt.

In unserer letzten Installation auf der Ars Electronica 1990 "Die imaginäre Bibliothek"⁵ haben wir versucht, eine Umgebung für eine kollektive Textproduktion zu schaffen.

Zwei Computer-Terminals mit einer HyperText-Oberfläche sind innerhalb eines Rundbaus inmitten von Büchern und Buch-Objekten plazierte. Ausgewählte Exemplare, die verschiedene Ausprägungen nicht-linearer Buchformen vorstellen: zeilenweises Blättern, kombinatorisches Lesen, visuelle Poesie...

¹ Europäisches Medienkunstfestival Osnabrück 88 und 89, Ars Electronica 89 und 90

² z.B. in Kooperation mit Arnold Dreyblatt für seine HyperText-Performance "Who is Who in Central & East Europe", Berlin 1991

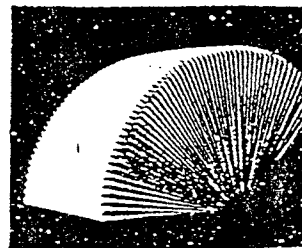
³ Idensen, Heiko/Krohn, Matthias, Connect it! in: Ars Electronica (Hrsg.), Im Netz der Systeme, Berlin (Merve) 1990, S.123-140
Idensen/Krohn, Vom Hypertext in der Kunst zur Kunst des Hypertext, in: Gloor/Streiz (Hrsg.), Hypertext und Hypermedia, Berlin 1990. S.292-300

Idensen/Krohn, Kunst-Netzwerke: Ideen als Objekte, in: Rötzer, Florian, Digitaler Schein, Frankfurt/Main 1991, S. 365-390 -

⁴ Michel Butor, Orte - Okopenko, A., Lexikonroman - Jorge Luis Borges, Labyrinth - Raymond Roussel, Nouvelle Impressions d'Afrique - Walter Abish, Abbild, Anspruch, Absicht. in: Quer durch das große Nichts - Raymond Quenau, Hunderttausend Milliarden Gedichte - Arno Schmidt, Zettels Traum - Michel Leiris, Aurora - Apollinaires Kalligramme - Diderots ENZYKLOPÄDIE und Benjamins PASSAGENWERK - Marvin Minsky, Mentopolis - Visuelle Poesie, Künstler- und Objektbücher - Fluxus Aktionen und Happenings -

⁵ Idensen/Krohn, Die imaginäre Bibliothek, in: Hattinger/Weibel (Hg.), Digitale Träume, Linz 1990

Während die Leser als Reisende in einer Bibliothek mit antiken und modernen Autoren kommunizieren, drucken zwei Drucker im Hintergrund permanent die Lese-Touren der Benutzer aus. Diese Endlos-Ausdrucke -zu Buch-Rollen gerollt- vergrößern den Bestand der imaginären Bibliothek sichtlich. Die Arbeit der Leser wird Bestandteil der Bibliothek.



Das Ziel der Anwendung ist es, durch verzweigtes assoziatives Lesen und Navigieren den Benutzer in ein Netzwerk aus Texten zu verstricken. Aktives Lesen, das nach und nach in ein Schreiben übergeführt wird. Neben provokativen Zwischenbemerkungen, Fragen und Aufforderungen eines im Hintergrund lauernden Dämons wird der Leser durch assoziative Verweise von einem Text zum anderen geführt, von einem Buch zum anderen, zu korrespondierenden Bildern oder zu "generativen Büchern". Dies sind programmierte Bücher, in denen bestimmte text-generative Prinzipien als Produktionsanweisung umgesetzt sind, so daß der Leser selbst mittels dieser Programme poetische Prinzipien anwenden kann: Permutationen, Reihungen, Frage-Antwort-Komplexe, Mesostics, Romananfänge weiter-schreiben, visuelle Poesie, Texte aus Teilen vorgegebener Texte zusammenschneiden ...

Bisher hat die Technisierung des Schreibens und Lesens wenig dazu beigetragen, emphatische Konzepte und wunderbare Theorien offenerer literarischer Produktionsweisen in kulturelle Praxis umzusetzen. Ebenso wenig wie die Verbreitung von Tonband, Walkman, Photo-Kamera und Videoequipment alle Konsumenten in Reporter, Tonbandkünstler, Fotografen und Filmhersteller verwandeln konnte, hat der Computer massenhaft programmierende User oder 'schreibende Leser' im Netz der Aufschreibesysteme hervorgebracht.

Das ist verwunderlich, denn die poetischen Operationen mit denen Pound, Mallarmé, Joyce u.a. die Verwendung der Sprache revolutionieren sind denen ähnlich, die die Pioniere einer vernetzten Ideenproduktion in den 60er Jahren auf der neuen Wunschmaschine Computer implementieren und so für jedermann verfügbar machen:

- assoziativer Zugriff auf Daten unterschiedlichster Art: Alles kann mit allem verbunden werden! Poetische Verschiebungen: von der klassischen metaphorischen Reihung zum metonymischen Gleiten
- offene Texte, die an jeder Stelle verändert, ergänzt und mit anderen Textstellen (Bildern, Sequenzen) verknüpft werden können
- jedes Wort wird zu einem Knoten von Bedeutungen: möglicher Absprungort für neue Konstellationen, Anspielungen und Verweise
- Navigieren als Tätigkeit eines gezielten Umherschweifens vereinigt Operationen von Schreiben und Lesen
- die entstehenden Strukturen aus einer Vielzahl von Querverweisen sind Texten analog, in denen mehrere Stimmen zusammenklingen: Intertext
- gleichzeitiger Zugriff einer Vielzahl unterschiedlicher Benutzer eröffnet einen offenen Diskursraum als Grundlage für kollektive nicht-hierarchische Wissensproduktion
- Wenn das Fragment (enzyklopädische Wissensbausteine mit aktiver Vernetzungsmöglichkeit durch den Leser als Reste der 'großen Erzählungen' - Produktionswissen wie

Tabellenkalkulationen, Baumstrukturen, Genealogien, Fahrpläne, Telefonnummern, Terminplanungen, Literatur- und Zitatdatenbanken...) modern ist und der Essay (als querlaufende Struktur rhizomatisch wuchernde Schwärme von Einzelheiten, Assoziationen, alltäglichen Erlebnissen, Gedanken, Randbemerkungen, Ideen, die sich aus den Bewegungen und Verknüpfungen der persönlichen Geschichten ergeben...) post-modern, dann ist HyperMedia der Beginn einer Kulturtechnik, die beides miteinander verbindet: Prozeß und Struktur verweben sich in der Poetik einer fortdauernden Vernetzung.

- **Visualisierung:** Hypermedia erzeugt immaterielle Hybride zwischen Text/Bild/gesprochener Sprache, in denen audiovisuelle Bearbeitungsweisen wie Schnitt und Montage neue Kompositiosmöglichkeiten eröffnen

- **Topographie:** ein Schreiben an Plätzen korrenpondiert mit Operationsformen antiker Mnemotechniken: Der HyperPoet durchstreift die Architektur seiner Ideenobjekte und generiert dabei poetische Strukturen, indem er eine Landkarte der Beziehungen zwischen Elementen anlegt

- **Spekulieren mit Bildern:** in einer Fortführung der Poetik visueller Poesie wird die bildliche Darstellung der Schrift weiterentwickelt: IKON-IDEOGRAMM-DIAGRAMM-ALPHABET-HYPertext

- **Synchronisation/Korrespondenzen:** die Synchronisation verschiedener nicht-linearer Datenströme auf der Oberfläche eines HyperMediums, das einen interaktiven Zugriff eines individuellen Benutzers quer zu feststehenden Präsentationsstrukturen ermöglicht, erzeugt ästhetische Sensationen multimedialer Ausprägung (analog zu Oper, Film, Live-Schaltungen) mit individuellen Zugriffsstrukturen (analog zum romantischen Leser), durch die der Benutzer selbst die unterschiedlichen Ausdrucksmomente zusammenführt, komponiert, steuert.

Diese Dialektik zwischen den HyperMedia Produktionen und den Benutzern/Interpreten verlangt nach komplexen strukturellen Polyphonien, nach der Gestaltung von Möglichkeitsfeldern, Poetiken des Andeutens, Kunstwerken in Bewegung.

"Gestützt auf Simulationsinstrumente (persönliche Metamedien), stellen wir **MODELLE ALTERNATIVER WIRKLICHKEITEN** her (Kunst); gestützt auf konversationelle Netzwerke (die öffentlichen Metamedien also), können wir aber auch die kulturellen Kontexte kontrollieren, die die Publikation und den Empfang dieser Modelle determinieren (Politik). Die Kontrolle des Kontextes beinhaltet die Kontrolle der Bedeutung, die Kontrolle der Bedeutung ist identisch mit der Kontrolle der Wirklichkeit."⁶

⁶ Gene Youngblood, Metadesign, in Kunstforum Bd.98, 2/89, Ästhetik des Immateriellen (Hg.Florian Rötzer), S.80

Vector spaces as the basic component of interactive systems. Towards a computer semiotics¹.

Peter Bøgh Andersen

*Department of Information and Media Science, University of Aarhus, Niels Juelsgade
84, DK-8200 Århus, Denmark. Phone 945 86 13 67 11*

ABSTRACT: The practical point of departure is design and implementation of an authoring system for interactive multimedia applications. My focus is on the aesthetic aspects, ranging from graphical display to narratives structures. In particular, I am interested in developing a theory of narration that exploits the special dynamic characteristics of computer systems, namely that the reader can interact with the system and the system can act itself. I suggest that a vector space consisting of objects and vectors can form the basis for such a theory; its basic narrative unit is the vector, a force emanating from an object that either attracts other objects or repels them. A good interactive multimedia product will contain sets of vector spaces, each with a simple internal structure but with the ability to generate a richness of events, most of which may not even have occurred to the author. I present arguments in favor of the vector in the domain of interactive narratives, but suggest that the concept has far wider applications: existing interfaces can be understood as vector spaces, and at least some algorithms lend themselves to this interpretation too. A final speculative chapter identifies scientific theories that can serve as a theoretical basis for the ideas: most obvious is semiotic interpretations of René Thom's catastrophe theory but the mathematics of fractals seems also interesting.

KEYWORDS: interface, interaction, interactive fiction, multimedia, object oriented programming, computers as media, programming methods, semiotics, catastrophe theory.

PUBLICATION: submitted to the workshop on "Hypersystem-Konzepte in Medien und kultureller Produktion", Lüneburg 15 - 17 July, 1991.

Introduction

The purpose of this paper is to reinterpret computer systems as media in which we can tell stories about human feelings, desires, aspirations, and conflicts in new ways. The paper contains both practical and theoretical issues. Sections 1 to 2 search for basic elements of expression that are as characteristic of the computer medium as the *montage* techniques are of film. It has a practical basis in design of an editor for building interactive multimedia systems. Section 3 generalizes the findings to apply to other interactive interfaces. Section 4 sketches the theoretical background for interpreting computer systems as media and sign systems. On this background, section 5 discusses the relevance of the vector concept to programming. Finally, section 6 widens the theoretical scope one step further in placing the ideas of the paper in the setting of catastrophe theory.

I have tried to write the paper for readers with semiotic as well as computer science backgrounds, so I sometimes have to give explanations and definitions that are trivial for one of the groups.

¹ Thanks to Per Aage Brandt for good discussions on catastrophe theory.

1. Computers as media

Many computer systems are used as media in the sense that their primary function is to disseminate and present information: mail and conference systems are obvious analogues to paper mail and telephone, and hypertext and multimedia applications are used as presentations, textbooks, and public information systems. Videogames — the entertainment variant of computer systems — have been around for many years.

The purpose of these applications is to tell a story, but most of them borrow too heavily from the media they are replacing: the book, the picture, and the film. The interactive possibilities often degenerate to clicking buttons or parts of text and pictures, which is just a sophisticated mode of turning pages in a book. The style and composition of these products are often undeveloped resulting in the well-known problems of fragmentary readings and getting lost in the system, so sometimes one wonders what are the benefits of interactive media really.

Interactive media have the same problem as film had in the beginning of the century; for over twenty years, many silent movies were just filmed theater that could only compete with real theater on ticket price. Only when the special aesthetic nature of the film was discovered, namely the technique of montage, did the film rise from cheap entertainment to one of the most powerful art forms in this century. The interactive media are just as much in want of a suitable aesthetics that will sever its links to tradition and turn it into a new medium in its own right.

In the first part of this paper I shall describe some ideas one of my colleagues, Berit Holmqvist, and I got when we tried to make interactive fiction. Our purpose was to invent narrative techniques for interactive systems, and we thought that fiction would be a good genre to start with because of the freedom and experimentation it traditionally encourages.

2. Vector spaces in interactive narratives

2.1. Introduction

The first product we made² was an introduction to an interactive story³. The story was mostly an exercise in transferring pictorial and filmic techniques to the screen: use of lights, experiments with multiple windows, exploring the possibilities of animation and finding out which kind of pictures look nice on the screen.

The introduction was a slide-show, but — we said to ourselves — the real story would of course be interactive. When we started work on it, we discovered that we had not touched the real problems yet. No matter what we did we always ended up with an essentially linear story that could branch in different directions: first we see two pictures; then we can click at one of three buttons. If we click the left button we see three pictures, and are again brought to a branching point, etc., etc. Instead of selling one book, we were selling ten books in one cover, but what was the point of that?

² Andersen & Holmqvist 1990a, 1990b.

³ On interactive fiction, see Bates (no year), Bates & Joyce 1987, Bolter & Joyce 1987, Laurel 1986, 1989, Smith & Bates 1989, Moulthrop 1989. A textual example of an interactive fiction is M. Joyce: *Afternoon*. Eastgate Systems: Boston 1987.

Interaction was still divorced from the narrative. Clicking a button is a meta-action that is not part of the story, and even if the button is disguised as a tree in the landscape, it remained just as much "meta" — who has ever heard of a story in which tapping on a tree is an important and meaningful part of the narrative? Normally we require of good texts that there should be correspondence between contents and form. Our example story was a love-story involving feelings and sensuality; button-clicking certainly has no correspondence with this contents, and the result sometimes was unintentionally comic and bordered to bad taste.

One of our problems was that paper contaminates; writing on paper forces the thoughts into the Procrustes bed of linearity. Therefore we decided to write an editor for making interactive narratives, in order to get rid of the paper. But writing an editor for building interactive narratives is not just a technical task; in fact, it involves creating a theory for them. In such a theory you need to define the basic building blocks — the phonemes of interactive narratives —, you need to define which kind of wholes they can be combined into — the interactive syllables —, and you need to provide methods for combination — the syntax of narrative systems.

The basic theory of interactive narratives that emerged during this work is that the author should not see himself as writing a narrative but a world! This world is designed in such a way that there is a good chance for the reader to experience exciting or touching stories if he acts in this world. In Andersen & Holmqvist 1990b we call it a *narrative space*. This world is defined by simple rules, but the interaction of the rules should provide such richness that neither author nor reader can deduct the possible stories; the only way to get at them is to explore the world. Possible models for this world are cellular automata (e.g. the popular "Game of Life") and the theory of fractals.

On the background of this and other considerations, the basic building blocks must satisfy the following requirements:

- they must not be inherently sequential. Instead they should be interpretable as concurrent processes, but the author should be able to superimpose sequentiality when it is needed. One reason is that we want to reverse the linear default belonging to paper and film, so that linearity must be explicitly decided. Another reason is that most interesting events involve several actors performing simultaneously: just think about car chases or shoot-outs in American films. They cannot be thought of as a sequence of single actions.
- they must be easily interpretable as signifying human actions and feelings because that is what most stories are about.
- they must have a natural visual appearance involving a space — two or three dimensional — in which objects change location, shape, color, visibility, etc. The reason is that that is what we can do on a computer screen. In addition, their visual appearance must be beautiful — interacting should be an artistic experience in itself, because we want contents to correspond to form.
- the user's interaction must be embedded in them, and not be superimposed from outside.
- they must be usable as a basis for programming, since the system will ultimately be controlled by a program. It means that they must be formalizable.

2.2. Basic definition of vectors

After several trials and errors we ended up with the concept of a *vector field*. Basically, a vector field consists of a collection of *objects* that attract and repel each other. These forces are described as *vectors*. Vectors must have a *destination* which does the attraction or repulsion and they must specify the *subject(s)* that is subject to these forces. Attraction and repulsion have straightforward visual displays, since the subject is moved towards or away from the destination, depending upon the direction of the vector.

A chase can be expressed by two vectors; the first one is positive and signifies the pursuer's desire to get hold of the pursued, so its subject will be the pursuer and its destination the pursued. The other one will be a negative vector from the pursued to the pursuer.

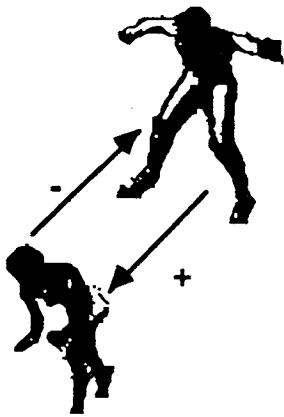


Fig. 1. A chase.

Ambiguous feelings can be implemented by two vectors between the experiencer and the object of experience. One will be negative, the other positive. The experiencer will both be attracted to and repelled from the object. This can be displayed as in ballet by the experiencer "dancing" to and from the value-laden object.

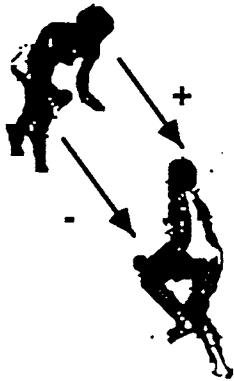


Fig. 2. Ambiguous feelings.

Topological properties of the scene can also be signified by vectors. In this case they will not hold between actors, but between actors and areas. A steep cliff for example, must make the actors slide down if they are on the top, and act as an obstacle for actors on the ground.



Fig. 3. Topological vector.

The last example shows that we have to add the concept of a *range* to a vector. The vector works only when the subject is inside the range, so in this way we can specify that the gravity force works only on the slope, not on the ground. This range is also necessary in the first two examples, since we might like the chase to begin only after the two actors are sufficiently near so that they can discover each other. Also, the pursuer should have a chance of escaping. This is impossible if the range is indefinitely large. However,

if it has a limited range, the user might be able to help the pursued out of it, so that the pursuer abandons the hunt.

Vectors differ in the definition of what it means for an object to be inside the range. If the vector has a physical interpretation, we need several variations in order to maintain the physical illusion. In the cliff example, the vector should take effect when the feet of the actor intersects with the slope. Thus, an object is inside the range if a part of it intersects with the range. This is also the case in "normal" graphical interfaces that employ a dragging operation. A special part of the cursor — its "hotspot" — is the range (and destination), and the user can drag any object that intersects with the hotspot. Our system is mostly concerned with social vectors, so we use a simple general definition: an object is within a vector if its midpoint is within it.

In addition to the range, we would like to have a *center* of the vector in which it stops working, since when a person approaches another person, the former will stop at a certain distance to the other one. These "stopping" distances are socially coded and have a meaning: we will keep longer distance from strangers on the street than from interlocutors at a party. If the rules of proximity are violated, we feel uncomfortable.

The vector acts by pulling or pushing, but we will need to specify different *forces*: if two actors are attracted to each other, the speed of the movement will signify the strength of the attraction: slow movements signify a tentative contact-seeking, while fast movements will be interpreted as strong feelings. Also, *change* of forces are useful: a sequence where the protagonists start by moving slowly towards each other, then show their feelings, rushes towards each other and embrace, is a cliché in films.

But just moving a person towards another person is seldom sufficient since such movements can mean opposite things; sometimes it means the subject loves the destination, sometimes that the subject wants to catch and hurt the destination. Similarly, actors in the grip of the gravity vector of the slope above should not be sitting down eating lunch, but assume a posture people normally assume when they fall. In order for the vector to communicate its meaning, it must specify a *transformation* that the subject suffers while influenced by the vector.

The nature of the transformation depends upon the way the force influences the subject. If it applies uniformly to the whole object the object can undergo a series of global shape-changes — a "dance" —, for example a series of postures shown after each other like an animated comic strip. Since we use bitmaps for graphical display, this is the only transformation we can use. The same technique is found in graphical interfaces, where certain areas of the screen changes the cursor.

The vector can also apply unevenly to the subject, causing some kind of deformation of it by influencing only one of its parts. As shown below, vector graphics often uses this technique for modifying drawings. The drawing contains small ranges (square "handles") that can be pulled by means of the cursor. The cursor is the destination, and the graphical object is the subject. The square handles defines the part of the object that can be within the range of the vector (the domain); you cannot deform the graphic if the handle does not intersect with the tip of the arrow.

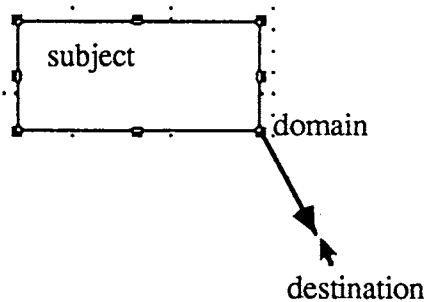


Fig. 4. Vector whose force pulls unevenly. The small arrow is not a vector but the cursor.

2.3. Tense and aspect

A vector needs a *tense* inflection since we want to be able to distinguish between *present tense* — the vector is working and the transformation is in effect — and two different *past* tenses: either it can end with *failure* or it can end with *success*. These two kinds of endings must also be signified by posture changes of subject and destination. If the pursued is caught, the pursuer should assume a triumphant shape, while the pursued must express anger or fright, while the pursued should express relief if the chase vector fails.

But tenses can be used for more than that. Until now, all vectors are assumed to apply concurrently whenever they can, but we can achieve partial linear structures if we add preconditions to vectors referring to the tense of other vectors. The idea is that a vector may only apply if the vectors mentioned in its preconditions have a certain tense. In the system there are four tenses:

- a future tense: a vector can only start its work if it is in the future tense. The first time it is activated, its tense becomes
- present tense: as long as the vector is in force, it stays in the present tense. A vector can stop being active for two reasons. Either the subject has arrived at the destination, or the destination has succeeded in moving outside the range of the vector. In the first case, we have
- perfect failure: it means that the vector has stopped but the final state is not reached; in the second case we have
- perfect success, meaning that the subject has arrived at the destination.

These concepts can be used to specify partially ordered plots. Here is a very simple soap opera involving the hero mr. White, the heroine miss Beauty-in-Distress, and the villain mr. Black.

1. After Black has caught Beauty-in-Distress (perfect success) he rapes her (present). Rape becomes present only when catch has become perfect success.
2. While Black is trying catch to Beauty-in-Distress (present), she will try to escape him (present). Escape becomes present when catching is present.

3. While Black is trying to rape Beauty-in-Distress (present), White will try to rescue her (present). Rescue becomes present when rape is present.
4. If Black has raped Beauty-in-Distress (perfect success), White will try to revenge her (present). Revenge becomes present when rape has become perfect success.

As it stands, mr. Black will stop his evil work if mr. White succeeds in preventing the rape. The rape vector has become perfect failure, and there is no way to make it future and thereby potential active again. Regrettably, villains do not give up that easy, so we must have a way of specifying if the vector becomes active after a failure or stays inactive. For this purpose, we can add an inflection for "Aktionsart" or "aspect" as it is sometimes called⁴.

Some vectors can be iterative in the sense that perfect success automatically turns into a future tense; a vector *get away* stays in the perfect after having succeeded and has therefore terminative aspect, but a vector like *stay away* is iterative in the sense that it is activated even after having succeeded. Maybe the vector *catch* and *rape* should be "try again vectors" that automatically become future after perfect failure, but stay in perfect success tense.

The reader might wonder why we use terms like tense and aspect. The reason we do it is not that we try to simulate language by means of pictorial codes; the reason is rather, that we want to support the readers' production of interpretants of the system. We view the objects and the processes on the screen as signs; according to Peirce, a sign stands for something for somebody in some respect. The sign itself, the representamen, is the objects and processes on the screen. They stand for a drama involving the three actors which are the objects of the signs. But signs cannot occur without an interpreter producing an interpretant; an interpretant is a new sign that stands for the object in a different way than the first sign did. In our case, the reader of the story cannot help producing interpretants when "reading" the play. An important part of these interpretants will have linguistic form, so we try to use semantic categories that have analogues in language. We want the system to be "mentionable", it must be possible to speak about it.

2.4. Vector "syntagms"

The same argument can be applied when we design complex patterns of vectors. A single vector will seldom convey an interesting meaning. In the embryonic story above, at least 5 vectors are involved: catch, rape, escape, rescue, revenge.

We need therefore facilities for grouping vectors into larger syntagms⁵, but how should the syntagm hierarchy look like? The way language formalizes force-concepts could be a good starting point for getting ideas for groupings. According to Talmy 1985, the smallest complex structure of forces seems to involve two actors and two vectors. The actors are called the agonist and the antagonist. The agonist is the main character and is described as having some immanent drive towards action or rest. The antagonist tries to oppose the agonist, either by making an immobile agonist move, or by pre-

⁴ On aktionsart and aspect, see e.g. Noreen 1904 or Vendler 1957, 1967.

⁵ In linguistics a syntagm is a standardized pattern of words that belong together, like *The green tree in A green tree stood in the meadow*. The English term is "phrase", but I prefer the term syntagm because it is not so closely associated to linguistic material. In this paper it denotes any standardized pattern of elements that are functionally dependent upon each other, or contract identical functional dependencies to other elements.

venting a mobile agonist from moving. The force of the antagonist can be stronger or weaker than that of the agonist. This gives us four main types:

	Agonist inherently resting	Agonist inherently moving
Antagonist weaker	1. The hero kept his balance in spite of the steep cliff (hero = agonist, cliff = antagonist)	2. The hero kept climbing in spite of the steep cliff. (hero = agonist, cliff = antagonist)
Antagonist stronger	3. The steep cliff made the hero fall down (hero = agonist, cliff = antagonist)	4. The villain prevented the heroine from escaping. (heroine = agonist, villain = antagonist)

Talmy 1985 gives the following graphical representation of the two cases where the agonist is inherently moving:

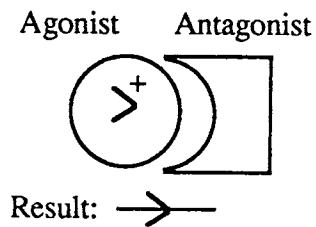


Fig.5. *The hero kept climbing in spite of the steep cliff.* + means: the strongest. > means tendency towards action, and • means tendency towards rest.

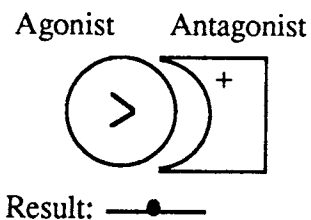


Fig. 6. *The villain prevented the heroine from escaping.*

They also are easy to represent in the framework presented so far:

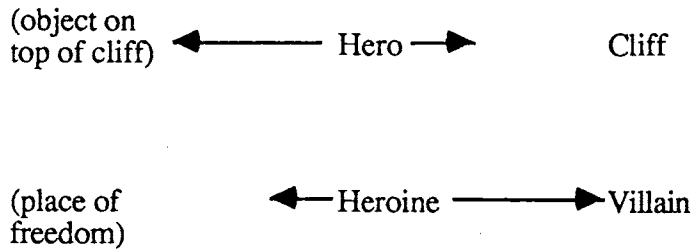


Fig. 7. Force dynamics with agonist inherently moving.

We have two actors, and one of the actors, the agonist, is influenced by two opposite vectors. The other two cases where the agonist is inherently resting are more difficult to represent. Number 1 would look like this in Talmay's notation

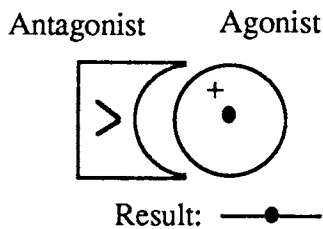


Fig. 8. *The hero kept his balance in spite of the steep cliff*

In order to represent this we could add a concept of inertia that in itself can prevent vectors to have effect. The problem is that inertia cannot be a property of the objects, since inertia can be very different for different kinds of vectors: although a person can very easily be moved emotionally, he may be so fat that he is difficult to move physically. Another possibility would be to add a vector whose subject and destination is the same person; such a vector could be interpreted as a vector from a future position to the present position. This vector would interact with the other vectors in the normal way: if it is positive and strong, it would cancel the forces of other vectors, making the actor immobile. If it is negative, it would signify an restless person, who cannot stay at the same place for one minute.

A third solution suggested in Brandt 1989 is to extend the dyad to a triad: a protagonist and two forces, the antagonist and the agonist, working on it. The inherent tendency would then be the result of the agonist influencing the protagonist. In the example *The hero kept his balance in spite of the steep cliff* the agonist is the friction between the hero and the cliff, while the antagonist is the gravity force.

Another thing to be learned from the force dynamics of language is that it clearly operates with "owners" of forces. In the examples above, the hero "owns" the climbing vector, the cliff the falling vector, the villain the preventing vector, and the heroine the escaping vector. In the present framework, ownership cannot be defined⁶. Sometimes the

⁶ On criteria for assigning actions to objects, see Bøgh Andersen 1990a, section II.1.2.1.

subject is seen as the owner (the villain trying to catch the heroine) while in other cases the destination can be the owner (the cliff dragging the hero down).

2.5. Displaying vectors

As said above, the system treats the active vectors as concurrent. The story itself progresses in a sequence of states; each state is generated from the previous one by calculating the effects of the active vectors on the objects, and displaying the results. The calculation can either be a vector sum or a vector product.

Suppose that the hero is dragged in two different directions:

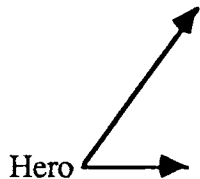


Fig. 9. Basic vector configuration.

Displaying the sum of the vectors would mean to move the hero along the diagonal (bold arrow) in the parallelogram below:

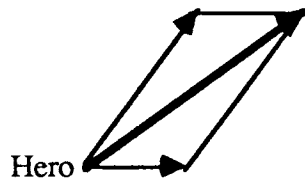


Fig. 10. Vector sum.

Displaying the product means to first move him along the first vector, and then in the next state along the second vector (bold lines shows the path he moves):

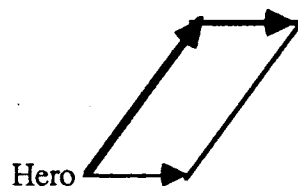


Fig. 11. Vector product.

The system supports both methods, because both have their uses. The product, for example, is useful, when a person is torn between conflicting vectors, e.g. duty and desire, whose sum amounts to immobility. A sum display would not communicate anything to the user, while the product will show the person oscillate first in the direction of duty, then back towards desire, then to duty again, etc.

2.6. Interaction

Now, what does user interaction look like in this framework? The basic principle is that the reader is an object that can influence the events in exactly the same way as the other objects, namely by being the destination of a vector. The reader is not in control; he does not act on the story, but must obey the internal logic of the narrative space. He can only influence the course of events under the same constraints as the fictive characters.

There are three methods for interaction:

1. the reader can try to pull an actor in a direction
2. the reader can show an actor something
3. the reader himself explore the scene of the action

1. Pulling. The reader can position the cursor on an object. When the mouse button is pressed, a transparent ghostlike shadow — called a *ghost* — of the object appear on the screen; this ghost, which acts as a cursor and should be interpreted as a possible future position of the object, can be dragged around on the screen until the button is released. The system then creates a vector pulling the object in the direction of the ghost. The transitory vector is the bold one below.

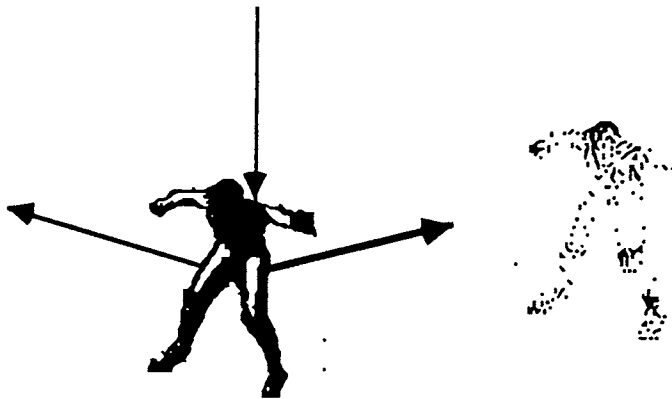


Fig. 12. Reader pulling actor.

But this vector must compete with all the other vectors influencing the object. In the example there is a vector pulling him to the left, and another pushing him down. The actual result of the user interaction is as usual the sum or product of all these vectors. It means, that in some cases the object will follow the reader's suggestions, in other cases the other forces are too strong, and the reader may only be able to delay the course of events — we have a "point of no return".

The relation between the cursor and the physical mouse can also be thought of as a vector; namely an indefinitely strong vector reaching from inside the machine out in the outer world. Its subject is the ghost cursor and its destination is the physical mouse. This means that the relation between software and external devices can also be understood within the vector framework. This is an advantage in multimedia systems, where we also want to control video discs, video recorders, and loudspeakers: the concepts of composition do not lose their validity when we cross the borders of the computer, so we can discuss use of peripherals in the same conceptual framework where issues of aesthetics

and composition are treated, thereby incorporating technical hardware discussions into the general aesthetic structure.

2. *Showing*. An animate actor can have a *view* associated with him. The view defines what he can see at the moment. Views are implemented as transparent colored spots resembling spots in the theater. The owner of a spot can see the objects inside the spot⁷. The spots can be moved by the reader who in this way can draw the actor's attention to new objects. The view of an actor influences the course of events by being a possible precondition of vectors along with preconditions pertaining to other vector's tense, cf. above. So a particular vector may only start its work if its subject sees specific objects. In the example above, we could add a view condition to the hero's rescue vector, so that it would only start working when the hero actually sees the heroine. In that case, it would be the reader that decides if she gets help.

Spots turned out to be a very beautiful means of interaction. The system uses color, and with a calculated color scheme, moving the spots can in itself be an artistic experience. Spots can be displayed as transparent pastel ovals discretely changing the color of the part of the screen it covers, but if most of the display is dark, use of inks can create the dramatic illusion of real spots highlighting parts of the screen and letting the rest remain in darkness. Moving a spot is then felt as exploring a dark room.

3. *Exploring*. Since we have already assigned a view to animate objects, the reader — being an object — has already got his own view for free. The reader's view is implemented as a window. He can see what is in the window at the moment, and therefore the story can be made to depend on what the reader actually sees now. The scene itself can be much larger than the the window and the reader moves his "spot" by dragging new parts of the scene into the window:

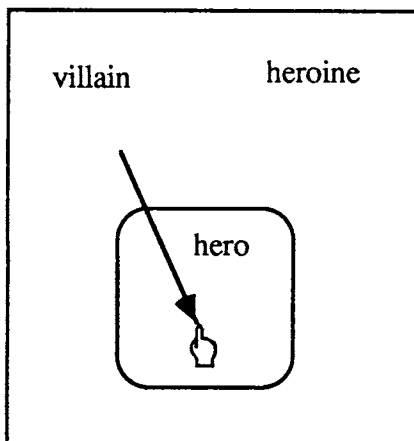


Fig. 13. Scene and window.

The large rectangle is the scene, and the round rectangle represents the window. In the example the scene contains the villain, the heroine, and the hero. The reader can only see those parts of the scene that are in the window, namely the hero. By holding the mouse button down and dragging, he can move the hidden part into view.

⁷ What is really needed is the ability of the spot owner to see a situation or an event — but implementing that requires some thinking.

Keeping check of the reader's view is a very good facility: else we risk that actors outside the window perform exiting and important actions that are never seen by the reader.

These are the ways the reader can interact with the story. He can be the destination of temporary vectors, and satisfy the view-preconditions for existing vectors, but can the reader himself be the subject of a vector? Since the reader is an object, the system architecture encourages such a notion, but how can it be implemented and what does it mean? Of course, the reader cannot be physically pulled and pushed, but a very usable implementation is to define vectors pulling and pushing the reader's spot. The effect would be that the scene sometimes begins to move by itself under the window. The interpretation is that now the author wants to show the reader something new, leading him on in the story.

This could help solving one of the basic problems in interactive fiction, namely the changed balance between reader and author. Since the story is interactive, the author has lost some of his old power and can no longer plan a long 600 pages novel. However, the solution is not supreme reader control where the reader can do what he wants, since in this case the reader has to write the story himself, and — because he is not a professional — will probably produce a lot of dangling ends and dull events. As a reader I would not take on that responsibility: I have spent good money on a product and do certainly not expect to do all the work myself.

A middle solution could be to define vectors gently influencing the reader's "spot". When the scene begins to move, I know that the author has something to tell me. I might choose to let him have his way, but may also resist and drag my "spot" (the window) in another direction.

2.7. Subclasses of vectors

In the preceding I have mainly treated one kind of vector, namely an *object vector* that holds between two objects.

However, we will need other types. The cliff in fig. 7 is an example of a different type of vector. In stead of having a destination in the shape of an object, we have an area inside which a force works in a uniform way. If objects are pulled with the same force in the same direction, their trajectory can be described as a simple linear equation, $y = kx$ with origin in the subject's present location. The force is then given by the equation plus an integer signifying the distance they travel in one step. However, we may want have more sophisticated topological properties: a circular space vector could exert a force defined by its tangents, and pits like the one below could be narratively very interesting:



Fig. 14. Narratively interesting pit.

The curve shows the trajectory the actor will follow when entering the space, and the force pulling him could be the sum of a vertical gravity vector and the "resistance" from the curve interpreted as a surface. As long as the actor is on the shallow parts of the pit, he

can still be rescued by the reader, but as he leaves the security of the "beach", the forces that drags him down grows stronger and stronger, and ends by dragging him down. So there is a narrative need for non-linear equations; the two minima above could also be useful: if the actor approaches the pit from the right, he is dragged down, but moving him further out into the danger now paradoxically turns out to be his rescue. The curve above has a more complicate equation; in fact it resembles one of the elementary catastrophes in catastrophe theory. It is called the cusp, and its equation is $y = x^4 + ax^2 + bx$. I shall elaborate on the similarities between the present framework and catastrophe theory in section 6.

Thus, in space vectors, the force specification is different than that in object vectors, and it makes no sense to have centers, directions and destinations. Still the two variants have many properties in common: views, vector preconditions, subjects, start posture, end posture, etc.

Some of the properties are identical but others are only similar. This is true of the concept of range. With objects, the intuition of a range is a circle around the object in which the force is effective, and since it is the object that is in the center of interest, the form of the range is not (very) relevant. Therefore, the range can be given as an integer denoting the radius of the circle.

Space vectors are different since they represent a force that is distributed in a specific area. The form of these areas is normally visible and significant: cliffs look different than pits and storms. All vectors must have the notion of a range, but the range is virtual in the class specification, and is differently specified in the different subclasses.

A third vector would be one that does not move graphics on at fixed background, but influences a sequence of frames. Let us call it a *sequence* vector.

It is convenient that there is a level in the program text where we are only concerned with the fact that we are dealing with a sequence of frames, but where we do not care whether the sequence is realized as a video tape, a video disc, a card sequence a la Apple's Hypercard, or an animation score as Micromind Director has it: if we want to mix these media in a multimedia application, it is an advantage if we could use the same concepts to control all these devices, and do not have to remember the special commands peculiar to each of them.

Like object vectors that have properties like start posture and end posture that denote graphical shapes the actor assumes when the vector begins and when it ends, sequence vectors have start and end properties too, but here they will denote the start frame and end frame of the segment to be played.

Their force is virtual, since the different realizations are different in that respect: Macromind Director offers a range from 1 to 60, while most videoplayers can only have a few speeds, e.g. normal, slowly, and fast. The positive/negative direction of the object vector could be reinterpreted as meaning forward/backward direction on the videotape.

In the actual implementation, we will need two vectors: one that pulls the start frame fast to the tape head without showing pictures on the monitor, and another one that moves the end frame to the playing head at playing speed.

Sometimes we will want the sequence to advance only at the request of the user; for example, if there is text to read, the reader should decide when he is ready to see the next frame. The verbal description of this, *The reader lets the "tape" move*, translates into a force dynamic pattern below consisting of two vectors: the Sequence has an inherent tendency to move its end frame towards the Monitor, while the Monitor itself successfully counteracts this movement, resulting in rest:

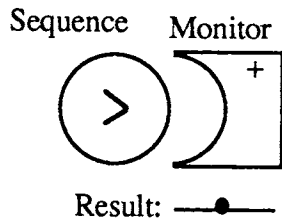


Fig. 15 . Tape sequence resting.

However, the user can temporarily remove the Monitor antagonist, thereby enabling the Sequence to move:

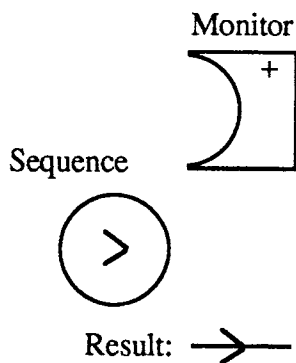


Fig. 16. Tape sequence moving.

The user's disabling of the Monitor force can be done in various ways, e.g. by pushing a button, or holding down a button.

The sequence vector will have subclasses with special properties. For example, the video subclass will have properties that are special to it: the baud rate of the signal, the port to use for communicating with the video hardware, etc.

The figure below shows a tentative class-hierarchy of the three vector-classes defined so far.

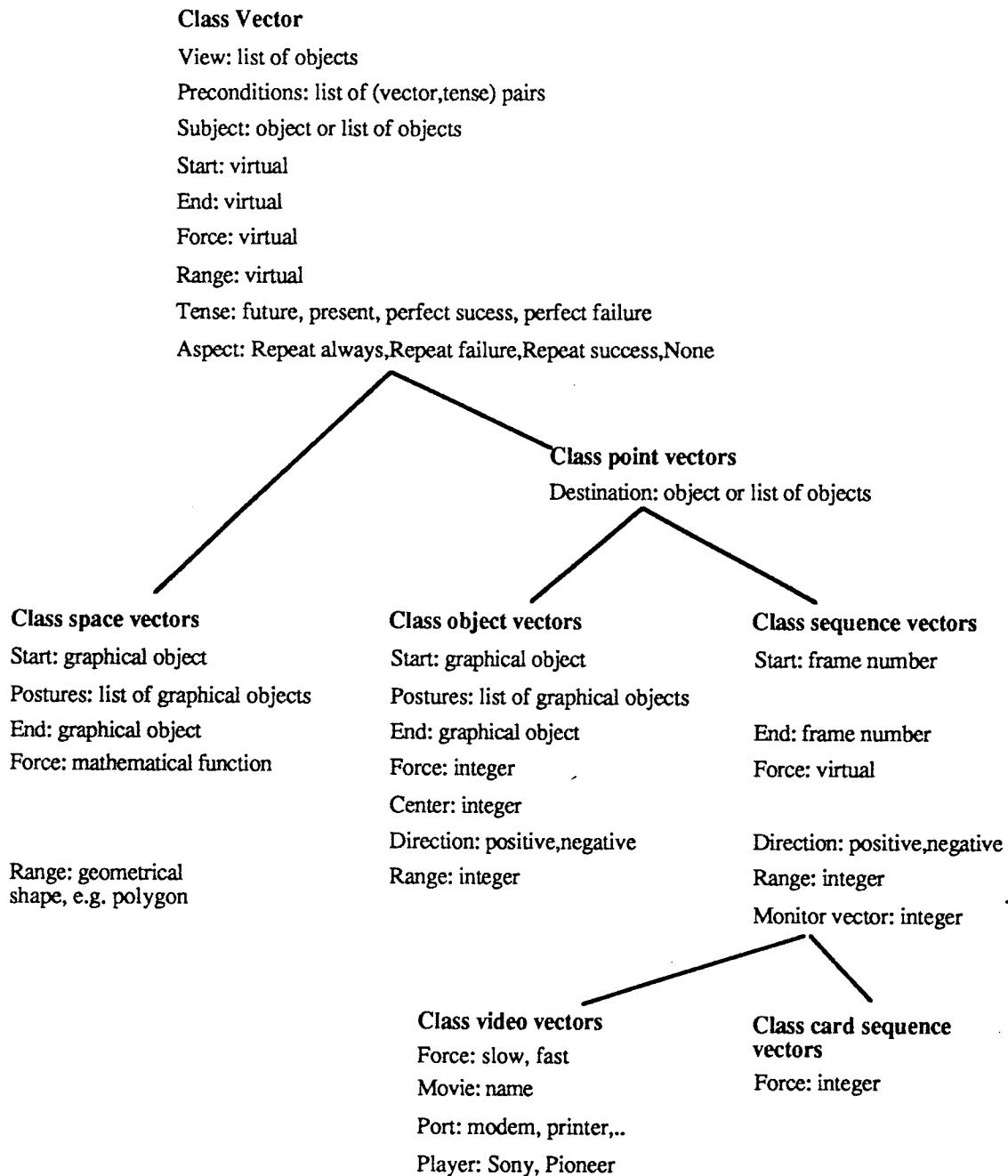


Fig. 17. Fragment of class-hierarchy for vectors. The convention in the diagram is that a class inherits all properties of the classes above it.

My examples have mostly been small pieces of fiction, but the tool can be useful to many "prose" applications, e.g. teaching systems. I am using the ideas in a museum system about the Scandinavian bronze age.

3. Vector spaces in interfaces

In this section I argue that the vector concepts are generally useful in design of graphical interactive interfaces. My main argument is simply that many existing interfaces lend themselves to a vector analysis.

As an example I shall use the interface to the Canvass drawing program reproduced below. I show how parts of the interface fits nicely into the framework.

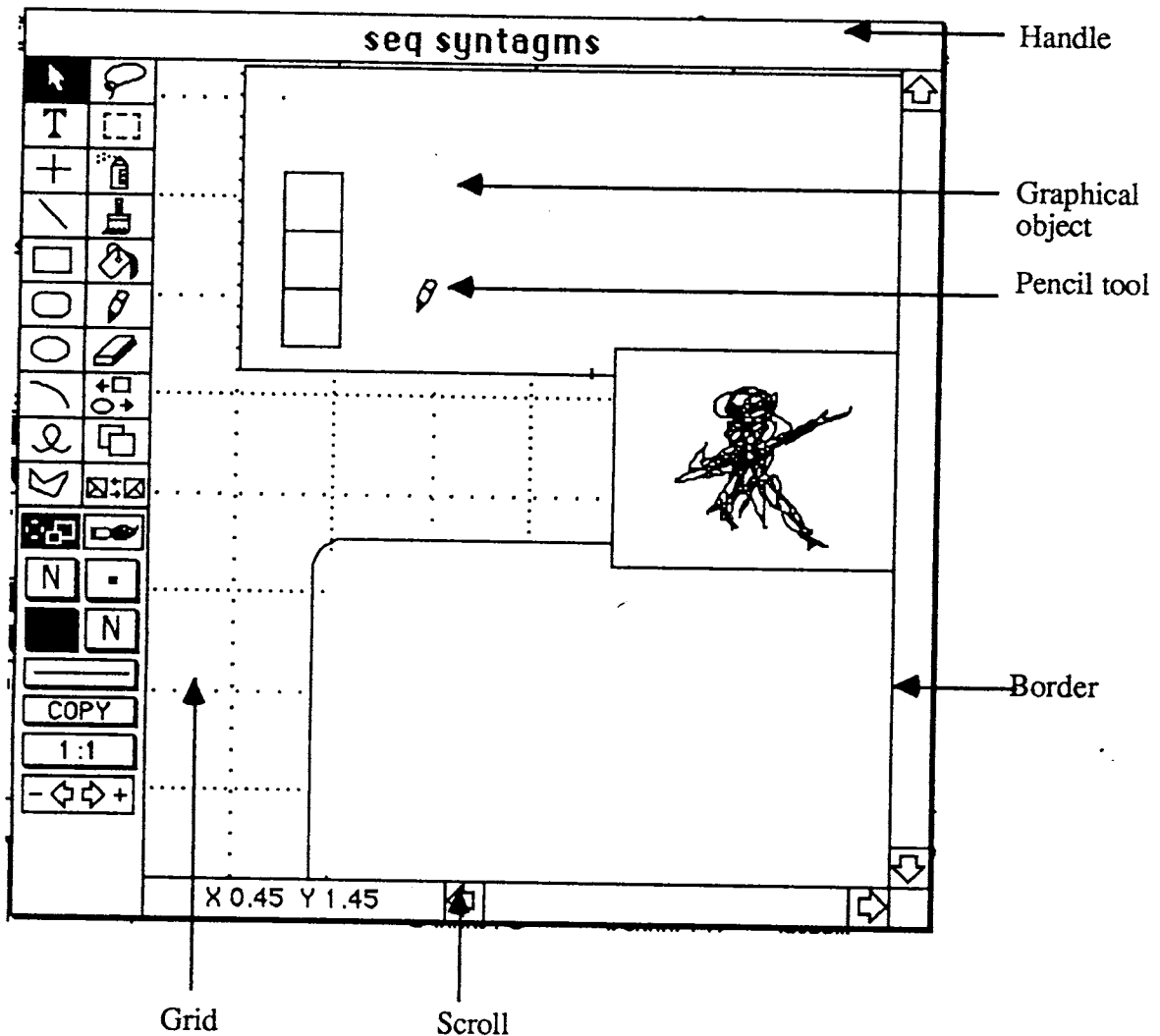


Fig. 18. Interface to the drawing program Canvass.

Grids are strong space vectors pulling graphical objects towards the gridlines. Their aspect is "repeat always" automatically turning a perfect tense into a future tense, since they are always active. Their range and force can be depicted as follows:

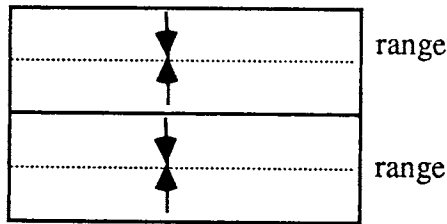


Fig. 19. Range and force of grid vectors.

Graphics can be dragged around on the canvass, but the dragging vector is opposed by the grid vector, so that the graphics are always pulled towards the grids.

The *borders* define space vectors that transform the cursor. The area enclosing the tool menu defines one vector that changes the cursor to an arrow, graphical objects change the arrow into a drawing tool, and the borders of the window is a third space vector that turns the cursor into an arrow.

The *scroll* arrows activate vectors that pull the window to the west, north, east or south, depending upon the arrow chosen.

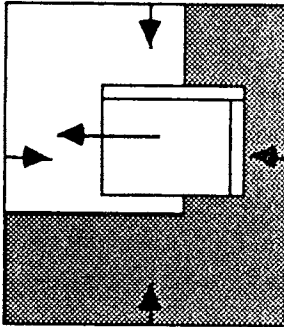


Fig. 20. Vectors involved in scroll-bar use.

The figure shows the vectors when we pull the window to the left. The background consists of a gray frame outside which the window cannot pass. On the gray frame is placed the paper canvass. When the window hits the borders of the gray frame, it is stopped by the opposing vectors emanating from the frame. The vector syntagm is identical to example 4 in section 2.4. A possible interpretant is:

The borders prevents the window from moving.

All graphical objects can be deformed in the manner described in section 2.2.

Except for the two space vectors, all vector ranges are attached to the subject, not to the destination.

These examples suggest that the vector concept is useful in general for designing graphical direct manipulation interfaces.

4. The media perspective on computer systems.

In the preceding I have described the concept of a vector space. The immediate practical motivation was design of an authoring tool for constructing interactive multimedia applications where aesthetics and narrative structure are important system properties.

I then argued that the concept is not confined to this kind of applications by showing how a class of existing interactive interfaces can be understood in these terms.

In section 5, I shall indicate how the vector concept can be applied to programming in general, but before I do that I present a broader theoretical framework within which the more practical suggestions make sense. A systematical exposition can be found in Bøgh Andersen 1990a.

4.1. Computer systems as sign systems

The underlying theory of the preceding is semiotics, the science of the life of signs in society.

From this point of view, a computer system is seen as a sign system and semiotics as a potential basis for actually programming computer systems⁸.

At first hearing this sounds implausible. After all, a computer is a machine, and we certainly do not want to use semiotics as a basis for constructing other kinds of machines like cars, vacuum-cleaners, and lawn-mowers. However, computer systems are not ordinary machines, assembled by means of bolts and screws. They are symbolic machines constructed and controlled by means of signs.

The interface of the systems is one obvious example of a computer based sign. A sign stands for something to somebody in some respect, and since the interface of a flight reservation system stands for flights and seats to the clerk, this interface is clearly a sign. Using the system involves interpretation and manipulation of text and pictures.

But underneath the interface, in the intestines of the system, we find also signs. The system itself is specified by a program text (that is a sign since it stands for the set of possible program execution to the programmer). The actual execution involves a compiler or interpreter that controls the computer by means of the program text, and since the compiler is a text standing for the set of permissible program texts, the compiler is also a sign — in fact it is a meta-sign that — in some versions — very much resembles an ordinary grammar.

Everything in a computer system, from top to bottom, is used as signs by some group of professionals. At each level there are texts that must be interpreted as statements or prescriptions about some present or future state of the system.

As we change level, the concepts signified by the texts change. On the lower levels, the meaning of the signs are related to the physical parts of the machine, like *registers* and *storage cells*. As we ascend, the texts are interpreted differently, we move away from a physical interpretation, and new software concepts appear, like *stacks*, *heaps*, and *variables*.

From this perspective, a computer system is nothing but a complex network of signs, and therefore every level contains aspects that can be treated semiotically. The word *aspects* is important; there are many aspects of programming, e.g. important issues of time and space requirements, that is completely outside a semiotic understanding. Semiotics is probably complementary to programming methods developed in a natural science context, since issues that can be treated systematically in one, is typically outside the range of the other.

⁸ On semiotic approaches to design and analysis of computer systems, see e.g. Rasmussen 1986, Nadin 1988, Bøgh Andersen 1990a, 199b, 1991.

4.2. The computer based sign.

The next question is naturally: what is a sign and what is a computer-based sign?

I use two variants of the sign-concept, namely the structuralist variant defined by Saussure and later elaborated by Hjelmslev and Eco, and the Peircian variant. The structuralist variant can be depicted as follows:

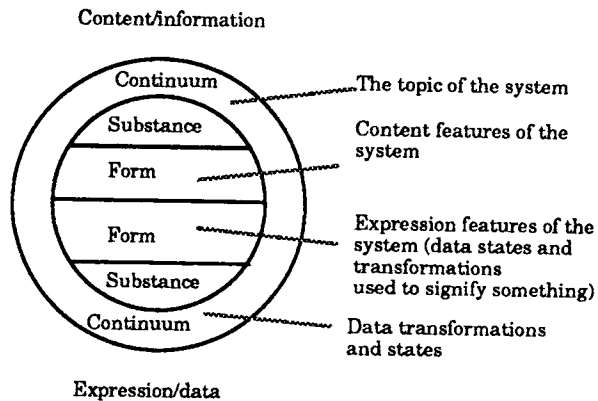


Fig. 21. The structuralist sign concept.

The structuralist sign is mainly a description of a sign *type*. It focuses on its social and institutional aspects. The sign has two main planes, the *expression* and the *content* plane, also called the signifier and the signified. Both planes have two aspects, a *substance* and a *form* aspect.

The substance of the sign is a part of the continuum articulated by the form of a sign occurrence. When a sign token is produced in some situation, two continua are articulated: significant distinctions are introduced into the expression continuum, e.g. sound in spoken language; these form elements are — via the sign function — correlated with semantic form elements, which in their turn establish distinctions in the content continuum.

One can envision the form/substance relation as a net (the form) casting its shadow on a undivided plane (the substance):

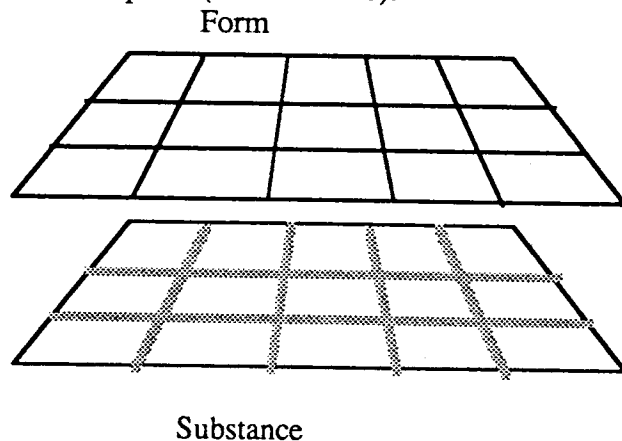


Fig. 22. Form and substance.

A flight reservation system can again serve as example of the content plane: the content continuum is airplanes, passengers, flights, departure and destination location and times. This continuum can be articulated in many ways, and a flight reservation system imposes quite special distinctions into it, reflecting the particular tasks of the clerk. For example, the number of motors of the plane is non-distinctive, whereas the number of seats is.

On the expression plane, the most obvious example is sound: each language imposes its own set of distinctions into the same sound continuum. For example, Danish has two [a]-sounds, /a/ and /æ/ which English lacks, so Danish has a borderline where English has an undivided surface.

However, we are not concerned with linguistic signs, but with computer-based signs. In fact, the theoretical purpose underlying the preceding sections is to propose a particular way of building computer-based signs. Our expression continuum is the set of all possible sequences of state-changes the machine can perform and the task is to impose distinctions on this continuum, differences that make a difference and convey meanings to the user.

The two planes of a sign are built out of smaller units. The smallest expression units in language are the phonemes. We can use them to build syllables and syllables can again be parts of words, which again are parts of tone-groups, utterances.

I suggest that vectors can be used as the smallest expression units of interactive computer-based signs, the computer analogy to phonemes⁹. A possible analogy to the syllable, consisting of two objects and two opposing vectors was described in section 2.4.

The vector concept imposes form on three kinds of continua: the continuum of distances (e.g. close vs far, contact vs non-contact), the continuum of movements (e.g. fast vs slow, towards vs away from) and the continuum of shapes (standing up vs lying down). Creating vector-based signs means to assign meanings to the form distinctions introduced or to combinations of form distinctions.

In addition, the tense and aspect system of the vector concept offers facilities for combining vectors into larger cohering syntagms. The tense of one vector (e.g. become present) can be made to depend upon the tense of another vector (e.g. being in the perfect tense).

But constructing these things does not automatically mean that we have constructed a sign, since users may not be able to interpret our product. Peirce's variant of the sign concept can help us clarify this point: in his framework it does not make sense to say that signs occur except when they are interpreted by some interpreter. Furthermore, in this process the interpreter cannot help reacting to the sign, producing a new sign called an *interpretant*. Therefore Peirce's sign contains three parts: the *representamen* (analogous to the expression plane), the *object* (analogous to the content plane), and the *interpretant* (which has no analogue in the structuralist sign concept). The following is mostly based on the structuralist concept, but I accept Peirce's analysis of the process of interpretation.

The interpretant, in particular, is important, since it is this concept that makes the present approach differ fundamentally from AI-approaches. If it is true that interpretation of

⁹ It is debatable how close the analogy is. For example, phonemes have no meaning in themselves, so they are not signs, but only building blocks for the expression plane of the morphemes, the smallest linguistic signs. It is not clear whether vectors can be said to have an inherent meaning. My own belief is that if they have it is very general, since moving an object to another object can have opposite interpretations, since the agent may either have negative or positive feelings towards the destination.

computer-based signs requires an interpretant to be produced, then an important requirement of the interface is simply that it enables production of an interpretant. In many cases this interpretant is verbal; this is obviously true in teaching situations where the interface must be explained to the novice, but it is also true in problem solving situations where a user talk about the system to colleagues. If we assume that our language to some extent structures our interpretation of the world, even when we do not speak aloud, verbal interpretants are also relevant when we just use the system silently. In order to make the system suitable for producing interpretants, we must analyze the interpretants, and see to it that the interface and interpretants are not incompatible. This means that we should use knowledge about the user's work language in systems design¹⁰.

The AI-way of using knowledge of language is to make the machine simulate it by designing a natural language interface and endow the machine with inference capabilities, but this strategy is a misunderstanding. In actual use there will by necessity be a human user that needs to interpret the system in order to do work, and it is the interpretant of this user we need to support. This means that the interface should not be conceived as a copy or simulation of the users' interpretant, but rather as resource for producing it.

The present approach is thus sceptic about positing similarities or identities between human mind and machines. For example, Turing's test must be considered a logical fallacy. The test says that if an observer is unable to tell the difference between the behavior of a machine and a human being, then we are entitled to say that the machine exhibits intelligence, but this deduction is inadmissible. What we do in Turing's test is to design an iconic computer-based sign standing for the human mind. However, in most other cases where we make icons of human reasoning and emotions, we certainly do not conclude that since the sign in one respect resembles its object, it must also resemble them in other aspects. For example, films are icons that stand for human cognitive processes, but we will never conclude that for this reason, there are deep-seated, non-trivial similarities between a film-projector or a celluloid strip and the human brain.

Another way of putting the same principles is the following:

Since a sign must consist of both signifier and signified, and since — according to Peirce — the signified only occurs when the sign actually occurs, is interpreted and an interpretant is produced, it is not correct to say that the programmer, via his program, *creates* signs. What he does is rather to *propose signs*, that in some circumstances become real signs (namely when users interpret them), but in many other circumstances never achieve this status, since they may not be seen by the users, and if seen, may be uninterpretable.

The essence of programming in a semiotic perspective is not to instruct the machine to do something; it is rather to use the machine to try to tell people something.

4.3. Views.

When we work with large computer systems shared by a whole organization, it is important to note that the sign systems are different in different department of the organization.

This is true both of the form and the substance aspect: the relevant content features can be organized very differently in the shop floor and in the manager's office, in the production plant and the accounting department. The substance articulated by the signs will typically also be different: on the shop floor the "continuum" articulated by the signs

¹⁰ See Holmqvist & Bøgh Andersen 1991.

consists of concrete work objects, tasks and workers, while the world of the manager consists of classes of work objects¹¹.

This is important since it tells us something about the sign structure of large shared systems. Since we must expect the same sign proposal to be interpreted differently in different departments — in fact, different departments may not be able to use the same sign proposal at all, but may require different signs —, the concept of different *views* on the same data acquires an important place in a semiotic systems analysis.

The figure below illustrates the basic vision of the computer system: a computer system is seen as a collection of data some of which can function as the expression plane of signs that emerge in the interpretative process. The contents of the signs depend upon the context in which the interpretation takes place: is it in the managers office or on the shop-floor?

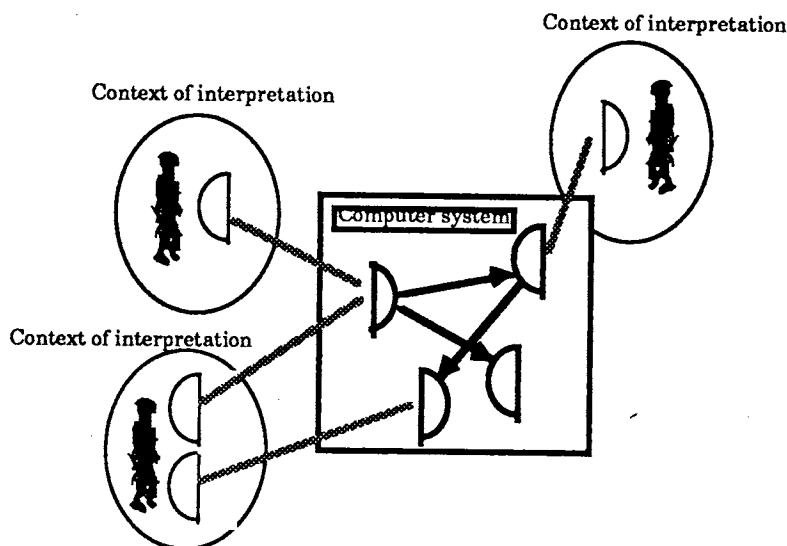


Fig. 23. The basic vision: the computer system viewed as a collection of "sign halves".

The structure of the views is closely related to the organizational structure — one can say that this technical aspect of the system is given an organizational interpretation. This is true also of the computer department responsible for developing and maintaining the system. They need special views in order to do their work properly, but the programming view is *in principle* not different from the other ones. Like other departments, they are shareholders in the system. The treatment of programming and program texts I give in the following is based on this assumption.

The sign-relation is symbolized by grey lines connecting the content and expression plane of the computer based signs. The black lines inside the computer systems represent formal relations between the data parts of the signs. The basic purpose of these formal relations is to enable the system to display different views and states of its object (in the Peircean sense) that people in the organization can make sense of and use as a basis for understanding and action. These formal relations are sometimes called a *model*; however, in the present approach the model has no value in itself. Its function is to ensure consistency between the views, so that information entered in one view shows up correctly in

¹¹ Holmqvist 1989.

other views. The model is a channel of communication between views. Different groups in the organization may need to view different selections out of the total collection of signs.

The organizational interpretation of the computer systems has definite consequences for the way we design systems. We immediately notice that people often change role in the organization and therefore need to change view of the system. For example, in our case the reader of interactive fiction may want to become an author.

If the views are unrelated, view-changing is hampered; in a library I investigated, customer service at the ground floor used one view of the library data while the cataloguing department on the second floor used another one. This made it difficult for librarians to change from customer service to cataloguing.

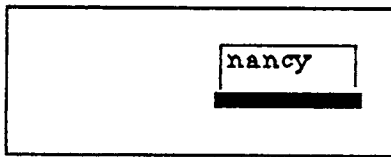
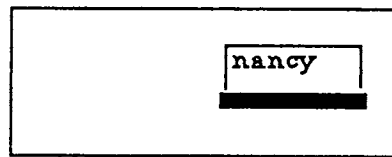
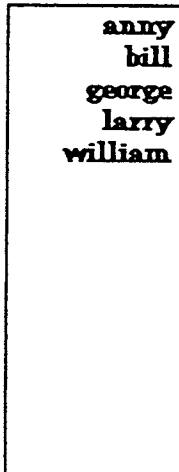
So although views must be different, a shared basic "vocabulary" for all views would be an advantage, because it enhances view-changes. The vector concept might be one possible candidate for such a "vocabulary" because vectors have a natural visual representation and because their basic spatial nature agrees well with the fact that space is a metaphor for computers users often adopt, cf. Bøgh Andersen 1990a, section I.2.4.

What remains to be explored is whether vectors can be used as a basic concept in programming. This is the topic of the next section.

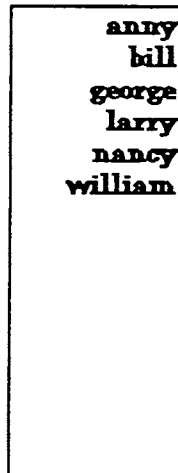
5. Vector spaces as algorithm specification.

I shall describe two kinds of views, the end-user view and a method-view. The latter specifies the general algorithmic method employed but does not give details of implementation. I use the binary search algorithm as example. In the framework outlined above, the difference between two views cannot be described as a distinction between the outer display (end-user view) of inner hidden processes (the method view). Rather, the two views are different simply because different stories are told in them.

Fig. 24 is a screen dump of a user view of the binary search algorithm. The user can type in a word and click the box enclosing the word. The word is then inserted alphabetically correct in the list called "Sorted List".

Binary search**Binary search****Sorted list**

Empty list

Sorted list

Empty list

Fig. 24. User view of the binary search algorithm.

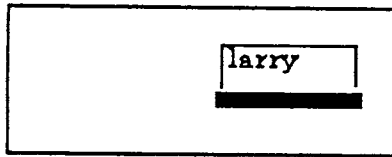
The meanings expressed in this view are very simple. The sequence of the elements in the list commutes and means "order" since if we shuffle the list elements around the new sequence has a different meaning, namely "disorder". In addition, the sequence, "word is not in list" followed by a mouseclick followed by "word is in list" has the meaning "the user inserts the word alphabetically in the list".

The story of the view is: the user clicks the box, and the word enters the list.

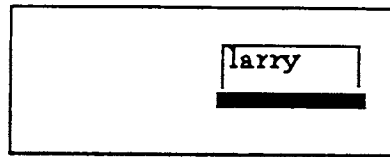
A vector interpretation at this level would be that the word has an internal propensity for action, but is hindered by the box. Clicking it disengages it from the box and lets it obey its internal vector.

This view does not tell *how* the word "enters the list". This is the topic of the method-view depicted below:

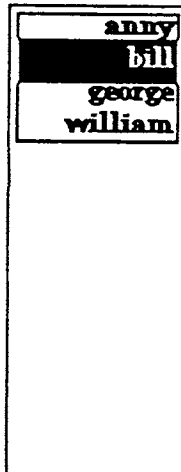
Binary search



Binary search

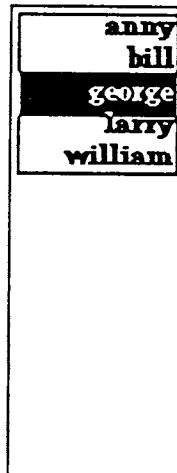


Sorted list



Empty list

Sorted list



Empty list

Fig. 25. Methods view to the binary search algorithm.

The method view consists of the same signs as before, but new signs have been added: the list has acquired two new components, the *area* (the rectangle) and the *marker* (the inverted line). The area of a list signifies the portion of the list where something can be inserted. In the beginning it encloses the whole list, and the purpose of the algorithm is to narrow the area down to one line where insertion must take place. The marker signifies the list element we are looking at at the moment.

New processes are also added: when the box is clicked, it moves to the left, bounces against the border of the large rectangle, and returns to its place. However, while it is away, the passage is open for the word that moves towards the list. In this way we implement the idea that clicking removes a hindrance.

The word is subject to two vectors: one that attracts it towards the midpoint of the area, and one attracting it towards the marker. The first one is in effect as long as the area is plastic, viz. can be shrunk, whereas the latter take effect when the area cannot be compressed any more.

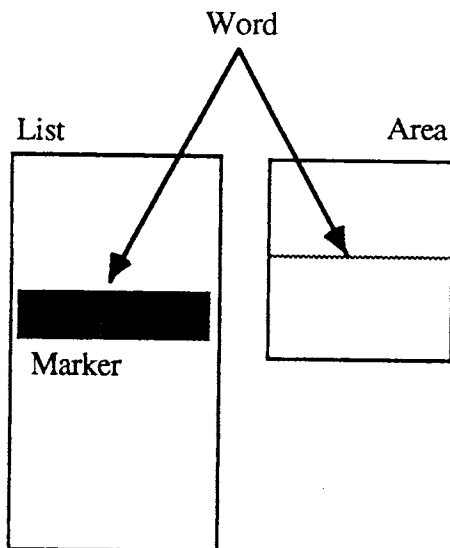


Fig. 26. The word as subject for vectors.

When the area has become rigid, the marker attracts the word, and once caught, the word is "ingested" by the list.

The word in its turn influences the area: if the word is alphabetically before the marker, it shrinks the bottom of the area halfway upwards, and if it is after the marker, it shrinks the top of the area halfway downwards.

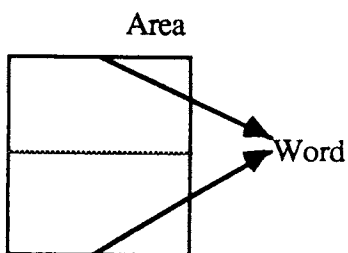


Fig. 27. The word as target for vectors.

The last component of the algorithm is a vector that drags the marker towards the midpoint of the area.

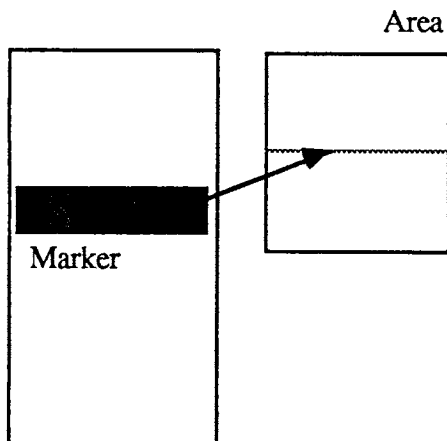


Fig. 28. The marker as target for vectors

If we combine the sign elements we get the following composite sign:

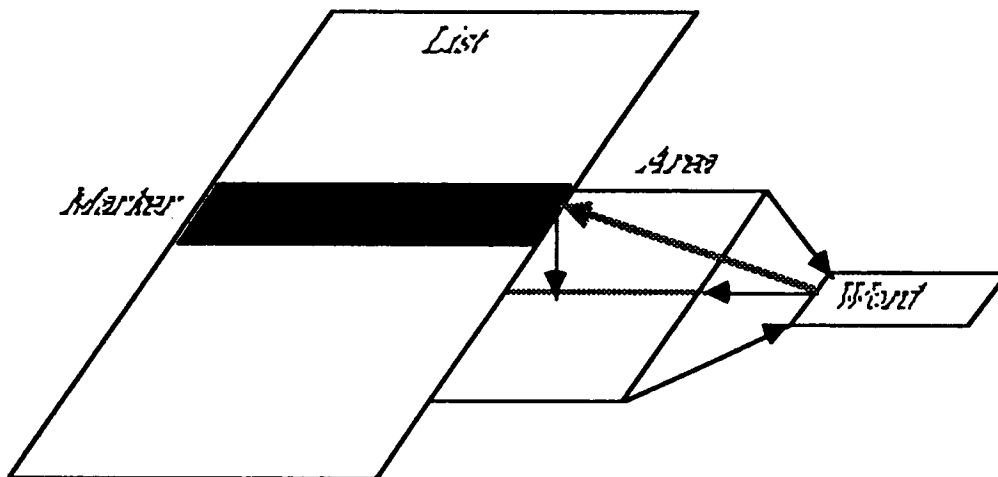


Fig. 29. Binary search as a topology of vectors.

This composite vector sign narrates the following story: the Marker is always attracted towards the middle of the Area, and the boundaries of the Area are attracted towards the Word: if the Word is larger than the Marker, then it pulls the Area's top towards it, whereas the bottom is pulled if the Word is smaller than the Marker. Both Word and Marker are pulled towards the middle of the Area (grey line), so when the Area shrinks both follow the new midpoint. When the Area becomes rigid (cannot be shrunk any more), the grey vector takes effect, pulling the Word to the Marker and inserting it.

In summary: in the beginning the area covers the whole list. The state is unstable and vectors push and pull, the word shrinking the area until it covers only one line. Then the word is inserted before or after the line and the state is stable. Thus we go from a state of *instability* to a state of *equilibrium*.

The program itself consists of a set of concurrently working vectors. An example of one of the vectors is the *move* vector:

```

if StateOf(bottom) = "absent" or StateOf(AWord) = "out" then
    moves AWord, MidpointOfArea (AList), out
end if

```

It has two preconditions: the bottom must either be absent or the word must already be out on adventure. The word subject is then pulled towards its destination, the midpoint of the list area. Its end state is "out".

The system visualizes all these movements in the space of the screen. Some people might just consider it a good tool for teaching algorithms, but I contend that the method view is not just a good illusion for teaching stupid students programming. It is neither an illusion nor a metaphor but stark reality in so far as we can talk about such notions in the realm of computers.

It is reality because the program text that creates the execution in fact describes it as a spatial force dynamics.

6. A catastrophe theoretical interpretation.

This last section contains a few speculative ideas motivated by the preceding sections.

As mentioned in section 4.2, a semiotic approach to interactive systems design must incorporate the notion of a substance that is formed. Catastrophe theory, originally developed by the French mathematician René Thom, is interesting in this connection because it is a mathematical theory about how form comes into existence.

One of the basic ideas of catastrophe theory comes from analysis of the behavior of certain equations called *elementary catastrophes* one of which, the cusp ($y = x^4 + ax^2 + bx$), was mentioned in section 2.7. With some values of a and b (the external control variables), the curve contains two minima, but if a and b is varied in certain ways, one of the minima gradually disappears. Thus, besides changing in a quantitative way, e.g. by increasing or decreasing minima, the cusp can also change in a qualitative way, e.g. from having two to only having one minimum. The latter change is called a catastrophe, and the hypothesis is that catastrophes can be used as a formalization of the notion of form.

The minima of the curves can be interpreted as attractors, the maxima as repulsors:

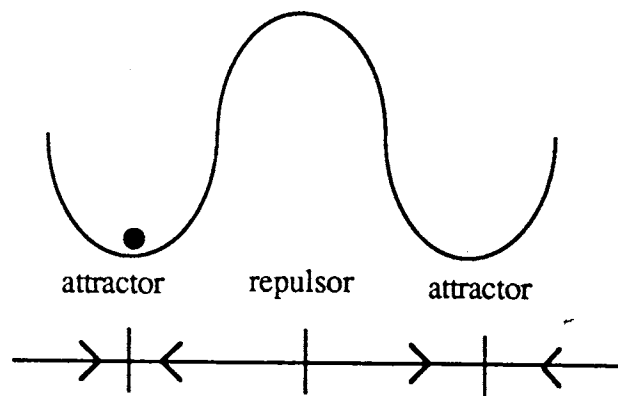


Fig. 30. The cusp with attractors and repulsors. The circle represents the system.

The easiest way of seeing this is to interpret the curve as a solid surface and adding a constant gravity force downward. The system, whose behavior we want to describe, is placed in this topology, and subjected to the combined effect of the gravity and the resistance of the "curve" surface. If the system is placed at a maximum, its state is unstable,

since it can and will roll down one of the slopes. On the other hand, if it is placed at a minimum, it will stay there, so its state is stable.

Catastrophe theory clearly distinguishes between two kinds of processes: processes in the internal topology and processes in the external topology. The former describe changes that occur in one particular topology, for example that the system rolls down to an attractor. The latter describes changes in the topology itself caused by the control variables. The former changes are changes of substance, the latter changes of form.

Several researchers have suggested a semiotic interpretation to these phenomena. (Petitot 1989, Brandt 1989a, 1989b, 1990a, 1990b). The *system* is translated as *actants* in the sense of French structuralism, e.g. the hero, the princess and the villain of the fairy tale. They are the narrative units that perform and suffer the actions of the story. The attractors and repulsors are properties the actors can assume or places they can be.

For example, in the beginning the princess is at home, but is influenced by the villain attractor; she is abducted and imprisoned in his cave. In order to bring her home, the topology has to change, so the hero now acts as an attractor, fights the villain, drags her out of the cave, and brings her home.

The basic structure of the narrative consists of *disturbances* (changes of the topology defining the equilibrium conditions) followed by adaptations of the actants to the new *equilibrium*.

This is precisely the stories you get in the system presented in the preceding, both in the soap opera and in the binary search example. The attractors and repulsors are identical to vectors; a story will consist of a sequence of *episodes*, each defined by a collection of vectors. This collection will strive towards an equilibrium state (if there is one which is of course not necessary). At some point the current episode is replaced by a new one, and since this is done by replacing one constellation of vectors by another one, it is a technically a catastrophe changing the topology of the narrative space¹².

I conclude with one last programming example that is described entirely in catastrophe theoretical terms, namely *recursion*.

My example is the Tower of Hanoi problem. You have three pegs and a number of discs of different sizes. All discs are on one peg. A disc is always placed on a larger one. You are required to move all discs to another peg obeying three rules:

1. a larger disc may not be placed on a smaller one
2. discs may not be placed on the ground
3. only one disc can be moved at a time

One solution to the algorithm is the following recursive procedure:

```

on MoveDiscs X,A,B,C
-- move X discs from A to C
  if X = 1 then
    -- simple case: move the one disc on A to C
    Propel TheTopOf(A),C
  else
    -- recursion step

```

¹² As a programming method, vector programming is similar to constraint oriented programming, cf. Freeman-Benson et al 1990.

```

    MoveDiscs X-1,A,C,B,
    MoveDiscs 1,A,∅,C
    MoveDiscs X-1,B,A,C
  end if
end MoveDiscs

```

We want to move X discs from A to C via B . But we *cannot* do that if there are more than one (cf. 3 above). So we first move $X-1$ discs from A to B via C , then we can move the one disc remaining, and finally move the $X-1$ discs from B to C via A .

Moving 3 discs from A to C via B is done by the call:

```

movediscs 3,A,B,C

```

Recursive methods normally have two parts:

1. the *simple case* that can immediately be performed
2. the *recursion step* where an impossible task is reformulated as a less impossible version of the same task.

The basic semantic field involved is modality: at present, the task is impossible, but we try to approach possibility (the simple case) by gradually making it less and less impossible (the recursion step).

If we plot the number of discs we try to move in a recursive call of the algorithm (viz. the X -variable) against time, we get the following curve:

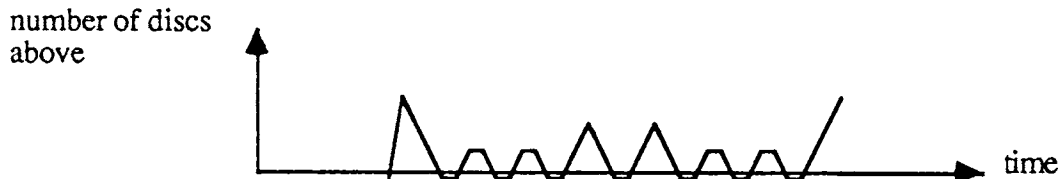


Fig. 31. Number of discs plotted against time.

The minima are the events where we go from impossibility to possibility (and further on to necessity because the method is deterministic), because only one disc is left and we just move it without further recursion. We can say that the machine moves in a modal space, starting well within the area of impossibility, stepwise moving towards necessity; when it enters necessity, it quickly returns into the realm of impossibility.

pulsor. According to the rules of the game, the protagonist (the ball) will roll (!) down in the antagonist minimum.

The middle part, with two attractor minima, represents the realm of possibility, since the ball can end in either of the two minima, movement or rest, while the lower part represents movement as necessary.

The meaning of the sentence itself is represented as a tour in this landscape: the term "because" presupposes that movement would not occur had the wind not blown, and leads to the prediction that the ball will come to a rest if the wind dies down. This little narrative behind the word is represented as follows: at first the ball is seen a shorter or longer time in the impossibility area, resting, then the topology changes to the lower necessity area because a wind rises, and the ball rolls down into the move-minimum in the internal topology.

In the Tower of Hanoi case, the negative of the number of discs can be interpreted as an external control variable, controlling the height of the rest antagonist; the larger the disc stack, the lower the minimum, and the stronger the rest antagonist.

Besides accomplishing the original purpose, to demonstrate how the vector concepts can be used as a programming concept, the benefits of this analysis might be:

- It could provide a classification of algorithms. The search program and the recursive Tower of Hanoi program are clearly different in that in the former case, the external topology does not change so that the whole story takes place in the internal topology striving towards equilibrium, while the latter story enfolds in the external topology, where one attractor grows weaker, and another stronger.
- We can clearly describe the difference between the two main parts of recursive algorithms: using the *simple case* part of the method is a change of form (a catastrophe, we go from impossibility to necessity, "we do a different thing"), while staying within the *recursive step* is a variation of substance (we continue doing the same thing, staying within impossibility).
- Compared to traditional semiotic analysis the framework gives us a concept of substance changes which is mandatory in programming since an algorithmic method is a method for forming a substance. We need concepts for what goes on between the confines of form.

Thus, it seems as if catastrophe theory can provide a theoretical background for the notion of interactive narratives described above. What started as a practical project turned out to have much wider theoretical and philosophical implications — which I think is a nice story.

References

- Andersen, P. Bøgh and B. Holmqvist. "Interactive Fiction. Artificial intelligence as a mode of sign production". *AI and Society* 4, 1990b.
- Andersen, P. Bøgh and B. Holmqvist. "Narrative computer systems. The dialectics of emotion and formalism". Paper presented at the conference *Computers and Writing III*, Edinburgh. 1990. Swedish translation in: J. F. Jensen (ed): *Computer-kultur — computer-medier — Computer-semiotik [Computer culture — computer media — computer semiotics]*. Nordic Summer University 1990a.
- Andersen, P. Bøgh and L. Mathiassen. "Systems development and use. A Science of the truth or a theory of lies". In: G. Bjerknes, P. Ehn and M. Kyng (eds.): *Computers and Democracy. A Scandinavian Challenge*. 395-419. Avebury, Aldershot. 1987.
- Andersen, P. Bøgh. "A semiotic approach to construction and assessment of computer systems". Invited paper to the conference "The Information Systems Arena of the 90s" IFIP WG 8.2 "The Interaction of Information Systems and the Organization. Copenhagen December 14 - 16th. In: Nissen et al (eds): *The Information Systems Arena of the 90s*. North Holland 1991.
- Andersen, P. Bøgh. *A Theory of Computer Semiotics. Semiotic Approaches to Construction and Assessment of Computer Systems*. Cambridge University Press. 1990a.
- Andersen, P. Bøgh. "Semiotics and informatics: computers as media". In: P. Ingwersen, L. Kajberg og A. Mark Pejtersen: *Information Technology and Information Use. Towards a Unified View of Information and Information Technology (64-97)*. London: Taylor Graham. 1986.
- Andersen, P. Bøgh. "Towards an aesthetics of hypertext systems. A semiotic approach". In A. Rizk et al (eds): *Hypertext: concepts, systems, and applications*. Cambridge University Press: Cambridge. 1990b.
- Bates, J. Oz Project. Overview and Schedule 1989-1992. School of Computer Science, Carnegie Mellon University: Pittsburg. (no year).
- Bolter, J. D. & M. Joyce. Hypertext and creative writing. *Hypertext '87 Proceedings*. 41-51. The ACM: New York. 1987.
- Brandt, P. Aa. "Quantity in iconicity. Esthetics and the Peircean sign." Charles Sanders Peirce Sesquicentennial Congress, 5-10 sept 1989b, Harvard University, Cambridge, Mass.
- Brandt, P. Aage. "Agonistique et analyse dynamique catastrophiste du modal et de l'aspectuel". *Semiotica*, vol. 77-1/3. 1989a.
- Brandt, P. Aage. "The dynamics of meaning. Three studies of modal semiotics". Copenhagen 1990a.
- Brandt, P. Aage. "The dynamics of modality: a catastrophe analysis". *RSSI* vol. 9, 1-2-3, 1990b.
- Eco, U. *A Theory of Semiotics*. London: The MacMillan Press. 1977.
- Eco, U. *Semiotics and the philosophy of language*. Bloomington: Indiana University Press. 1984.
- Freeman-Benson, B. N., J. Maloney & A. Borning. "An Incremental Constraint Solver". *Comm. of the ACM*, vol 33:1, 1990, pp. 54-63.
- Greimas, A.J. and J. Courtés. *Semiotique. Dictionnaire raisonné de la théorie du langage*. Paris: Hachette. 1979.
- Greimas, A.J. *Du Sens. Essais Sémiotique*. Paris: Éditions du Seuil. 1970.
- Greimas, A.J. *Sémantique Structurale*. Paris: Larousse. 1966.
- Hjelmslev, L. *Essais Linguistique*. Paris: Les Éditions de Minuit. 1971.
- Hjelmslev, L. *Prolegomena to a Theory of Language*. Menasha, Wisconsin: The University of Wisconsin Press. 1963. Translated from "Omkring Sprogteoriens Grundlæggelse". University of Copenhagen 1943, reprinted and published by Akademisk Forlag, Copenhagen 1966.

- Holmqvist, B. and G. Källgren. *Postgirot som Språkmiljö I (The Postal Giro as language environment)*. MINS 18. Department of Scandinavian Languages, University of Stockholm. 1986.
- Holmqvist, B. and P. Bøgh Andersen. "Language, perspective, and design". To appear in: J. Greenbaum and M. Kyng (eds.): *Design at Work*. Earlbaum. 1991.
- Holmqvist, B. and P. Bøgh Andersen. Work-language and information technology. *Journal of Pragmatics* 11: 327-357. 1987.
- Holmqvist, B. *Postgirot som Språkmiljö II (The Postal Giro as language environment)*. MINS 19. Department of Scandinavian Languages, University of Stockholm. 1986.
- Holmqvist, B. Work-language and perspective. In: *Scandinavian Journal of Information Systems*, no.1. University of Aalborg. 1989.
- Joyce, M. *Afternoon*. Eastgate Systems 1987.
- Laurel, B.K. *Interactive fantasy: a dramatic model*. Interactive arts lecture series. Carnegie-Mellon University: Pittsburg. 1989.
- Laurel, B.K. Interface as mimesis. In Norman, D.A. & S.W. Draper (eds.). *User Centered System Design*. Lawrence Earlbaum: Hillsdale, New Jersey. 67-86. 1986.
- Mathiassen, L. and P. Bøgh Andersen. "Semiotics and informatics: the impact of edp-based systems upon the professional language of nurses". In: van der Veer (ed): *Readings on Cognitive Ergonomics - Mind and Computers*. 226-247. Berlin: Springer-Verlag. 1984. Also in *Journal of Pragmatics* 10(1986): 1-26.
- Moulthrop, S. Hypertext and "the Hyperreal". *Hypertext '89 Proceedings*. 259-367. The ACM: New York. 1989.
- Nadin, M. "Interface design: A semiotic paradigm". *Semiotica* 69: 269-302. 1988.
- Noreen, A. 1904. *Vårt Språk [Our Language]*. Vol.5. Lund: C.W.K. Glerup.
- Ouellet, P. Semiotics, cognition, and artificial intelligence. Special issue of *Semiotica*. *Semiotica* 77. 1989.
- Petitot, J. "On the linguistic import of catastrophe theory". *Semiotica* 74-3/4. 179-209. 1989.
- Rasmussen, J. *Information processing and human-machine interaction*. North-Holland: New York. 1986.
- Smith, S. & J. Bates *Towards a theory of narrative for interactive fiction*. CMU-CS-89-121. School of Computer Science, Carnegie Mellon University: Pittsburg. 1989.
- Sweetser, Eve. "Root and epistemic modals. Causality in two worlds". *Berkeley Linguistic Society*, 8, Berkeley, Univ. of California. 1982.
- Talmay, L. "Force dynamics in language and thought. Parasession on Causatives and Subjectivity". Chicago Linguistic Society (21st Regional Meeting), Univ. Of Chicago. 1985.
- Vendler, Z. *Linguistics in Philosophy*. Ithaca, N.Y: Cornell University Press. 1967.
- Vendler, Z. "Verbs and times". *The Philosophical Review* . 143-160. 1957.

Ansätze eines hypermedialen Filmkatalogs

Frank Voeltz, Thomas Tesch, Tom Quellenberg, Ulf Hönisch

Arbeitsvorhaben HyFi¹

Universität Bremen, Studiengang Informatik

Filmgestaltung und Filmarchivierung gelten als eine Domäne zukünftiger Hypermedia-systeme [1] [2]. Der im Arbeitsvorhaben entwickelte rechnergestützte Filmkatalog stellt einen Versuch dar, Möglichkeiten und Probleme auf dem Weg dorthin zu erfassen, konzeptionelle Überlegungen zu entwickeln und Implementierungsanforderungen durch praktischen Systementwurf aufzuzeigen. Der Katalog integriert neben konventionellen Daten in Textform visuelle und akustische Information. Neben Problemen der Datenhaltung werden Fragen nach einem angemessenen Zugang zu diesem Material aufgeworfen. Dem Benutzer zur Suche in diesem Datenbestand effiziente Werkzeuge anzubieten, war ein Schwerpunkt des Systementwurfs.

Buch versus Rechner

Ein Katalog ist - und hier bildet der Filmkatalog keine Ausnahme - ein Verzeichnis. Die verzeichneten Gegenstände sind nach einem bestimmten System strukturiert. Da der Katalog gezielt bestimmte Information bereitstellen soll, müssen nicht nur diese Daten im Bestand enthalten sein, sondern ihre Struktur muß den zu erwartenden Anfragen angemessen sein. Ist sie es nicht, kann die gesuchte Information zwar gefunden werden, die Suche ist jedoch ineffizient. Die Suche einer Telefonnummer in einem nicht alphabetisch geordneten Verzeichnis erfordert entweder viel Zeit, den Rückgriff auf andere z.B. chronologische Kriterien oder schlicht eine glückliche Hand. Will man sich darauf nicht verlassen, muß eine ausreichend mächtige Struktur zur Organisation der Daten verwandt werden. Ein Verzeichnis kann nach (fast) beliebigen Kriterien strukturiert werden. Anliegen des Nutzers eines Filmkatalogs ist die Suche eines oder mehrerer Filme. Für die Erstellung des Katalogs bedeutet dies, daß die Autoren solche Kriterien festlegen müssen. Bei kleinen Datenbeständen und "einfachen" Daten bleibt der Sachverhalt überschaubar. Standardisierbare Anfragen tragen zur weiteren Vereinfachung bei.

Die Beschreibung eines einzelnen Films innerhalb des zugrunde gelegten Katalogs [3] umfaßt neben Angaben zu Filmtitel, Untertitel, Regisseur, Darsteller, Filmart (Dokumentarfilm, Spielfilm) und Filmtyp (Farbe, Schwarz-Weiß) noch ein Literaturverzeichnis und eine kurze Inhaltsangabe. Über die an einzelne Filme gebundenen Materialien hinaus bietet der Katalog Hinweise auf begleitende und vertiefende Materialien, die entlang der Schwerpunkte gegliedert sind. Verleihbedingungen ergänzen die Angaben. Standardisierte Fragen eines Benutzers, der einen bestimmten Film sucht oder ein bestimmtes Themengebiet behandeln möchte, werden durch die Gliederung des Inhaltsverzeichnisses nach Themenschwerpunkten und

¹ Das Arbeitsvorhaben Filmkatalog ist Teil des studentischen Lehrprojekts HYPERCORPUS im Studiengang Informatik der Universität Bremen. Das seit WS 1989 laufende zweijährige Projekt befaßt sich mit der Untersuchung und Umsetzung hypermedialer Systeme.

durch ein alphabetisches Register der Filmtitel unterstützt. Weitere Register, wie das der Regisseure sind ebenso denkbar, wie ein Schlagwortindex. So nützlich diese Hilfsmittel für den Benutzer sind, sie können immer nur einen Ausschnitt möglicher Strukturierungen erfassen und sind so zugleich beschränkende Restriktionen. Abweichende Anfragen sind zwar möglich, werden aber vom gewählten Aufbau der Daten nicht unterstützt. Die bloße Übertragung der Buchform auf den Rechner ermöglicht – speziell in großen Datenbeständen – lediglich eine schnellere Suche.

The screenshot displays a hypermedia interface for a film catalog. It features several overlapping windows with the following content:

- Inhalt** (Content):
 - Filmtitel: Athlet III – ein Schiff aber kein Dampfer
 - Filmtitel: Athlet III - ein Schiff aber kein Dampfer
 - Der Film beschreibt eine besondere Gruppe von Hafenerbeitern – die Besatzung des Schwimmkrans Athlet III. Die inhaltliche Schwerpunkte des Films sind: Beobachtungen des "Wir-Bewußtseins" einer im Unterschied zu sonstigen Arbeitsgruppen im Hafen ständig gleichbleibend zusammengesetzten Mannschaft, Beziehungsstrukturen, Kommunikationsformen, Organisations- und Planungsabläufe, Zusammenhang von Arbeits- und Lebenssituation, u.a.
- Literatur** (Literature):
 - Filmtitel: Athlet III – ein Schiff aber kein Dampfer
 - Abendroth, Michael / Beckenbech, Niels / Braun, Siegfried / Dombois, Rainer: Hafenerbeit. Eine industriesoziologische Untersuchung der Arbeits- und Betriebsverhältnisse in den bremischen Häfen. Campus Verlag 1979
 - Abendroth, Michael / Dombois, Rainer / Hessler, Helner: Vom Steuhecken zum Container. Eine vergleichende Untersuchung der tariflichen und betrieblichen Regelungen der Hafenerbeit in norddeutschen Häfen. Herausgegeben vom Hauptvorstand der Gewerkschaft Öffentliche Dienste, Transport und Verkehr. Januar
- Filmgruppe: Arbeits- und Lebenssituation von Werktätigen** (Film Group):
 - Arbeit** (Work): kein Dampfer
 - wirkende:** (acting):
 - Annes Holtussen
 - Bauer
 - Bevandenn
 - Busch
 - Falkenhain
 - Dieter Griesenbach
 - Herald Köster
 - Jürgen Lehmann
 - Jan Lehmann
- Schnitt:** (Editing):
 - Maxi Heinke
 - Hago von Zelindt
 - Warner Dobusch
- Untertitel:** (Subtitle): Die Mannschaft eines Schwimmkrans im Hafen von Bremen

Abb. 1: Daten im hypermedialen Filmkatalog.

Um die, durch die Bindung an den Rechner entstehenden Nachteile hinsichtlich Verfügbarkeit, Lesbarkeit und Ergonomie zu rechtfertigen, muß die Realisierung mehr sein, als eine Portierung der Buchform. Ein multimedialer Filmkatalog bedeutet daher zunächst ein Mehr an Filmdateien: Bilder des Filmtitels, von Schlüsselszenen und Darstellern, Bildsequenzen, die Filmmusik als Tonsequenzen, markante Dialoge. Diese Menge verlangt nach einer Strukturierung. Beziehungen können von einem Datum zu einer Vielzahl anderer Daten bestehen. Sie müssen explizit in der Datenmenge hergestellt werden. Diese Beziehungen sind aber nur in einem bestimmten Kontext interessant. Sie sind – mit vertretbarem Aufwand – nicht vollständig abbildbar. Bei der Integration eines neuen Datums müssen im Extremfall die Relationen zu allen bestehenden Daten geklärt werden.

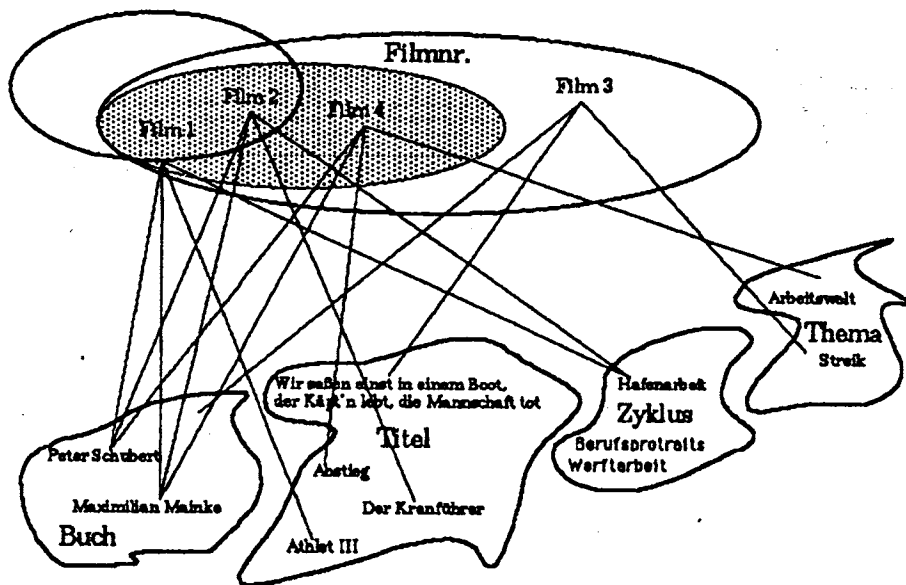


Abb. 2: Wechselnde Strukturen im Datenbestand

Eine vorgegebene Strukturierung ist unter diesen Umständen nicht mehr sinnvoll. Wechselnder Kontext bestimmt die aktuelle Strukturierung. Daher verlangt der Filmkatalog zum Aufbau geeigneterer Strukturen angemessene Werkzeuge. Der Benutzer kann die angebotene Filmmenge durch Suchbegriffe auf einen für ihn relevanten Ausschnitt begrenzen, wobei beliebige durch "und" sowie "oder" verknüpfte Begriffe zugelassen werden. Wiederholte Anfragen zur weiteren Einschränkung sind bei der Bearbeitung der Filmmenge ebenso von Interesse weil Rücksprungmöglichkeiten zu bereits realisierten, treffenderen Anfrageergebnissen. Suchen muß nicht auf textuelle Daten beschränkt bleiben, sondern ist über Hilfsstrukturen auf Bilder und Ton erweiterbar. Überschaubare Filmmengen können durch "Blättern" analog der Buchform erschlossen werden. Gewohnte Standardstrukturen des Filmbestands lassen sich bereits vom Autor durch vordefinierte, der Buchform entsprechende Anfragen zur Verfügung stellen.

Die Wahl des Datenbestands zu einem Film stellt weiterhin Restriktionen dar. Die Entbindung des Autors von der Aufgabe den Bestand zu strukturieren bietet dem Nutzer die Möglichkeit, diese Aufgabe mit den angebotenen Tools individuell zu bewerkstelligen. Dadurch werden differenziertere Anforderungen an den Filmkatalog zugelassen als sie von der heute üblichen Form unterstützt werden.

Konzept

Die Notwendigkeit angemessener Werkzeuge zur Aufbereitung und Strukturierung der im Katalog verwalteten Informationen bildete die wesentliche Grundlage, für das nachfolgend vorgestellte Konzept. Zwei für Hyper-Dokumente wesentliche Merkmale, die Integration verschiedener Datentypen und eine leistungsfähige Navigation im Hyperraum, wurden unter dem Gesichtspunkt der Erweiterbarkeit des Katalogs und einer "gemeinsamen" Behandlung der verschiedenen Datentypen betrachtet.

Dem Ziel der Integration diverser Datentypen entsprechend werden die drei Medientypen Text-, Ton- und Bildinformationen unterschieden. Damit läßt sich der Katalog vollständig gestalten, weil alle vorkommenden Informationen, wie z. B. Namen, Fließtexte oder Bilder, darauf abgebildet werden können. Zur effizienten Bearbeitung einzelner Daten genügt diese Grobgliederung nicht. Die Medientypen werden in eine erweiterbare Menge von Attributstypen mit Strings, Fließtext, Standbild, Filmsequenz, etc. gegliedert (Abb. 3). Für die Ausprägungen eines Attributstyps (Attribute) wurden generische Methoden entwickelt. Die Methode "Suchen" kann für alle Ausprägungen des Attributstyps Fließtext, wie Inhalt, Literatur, etc., verwendet werden (Abb. 3).

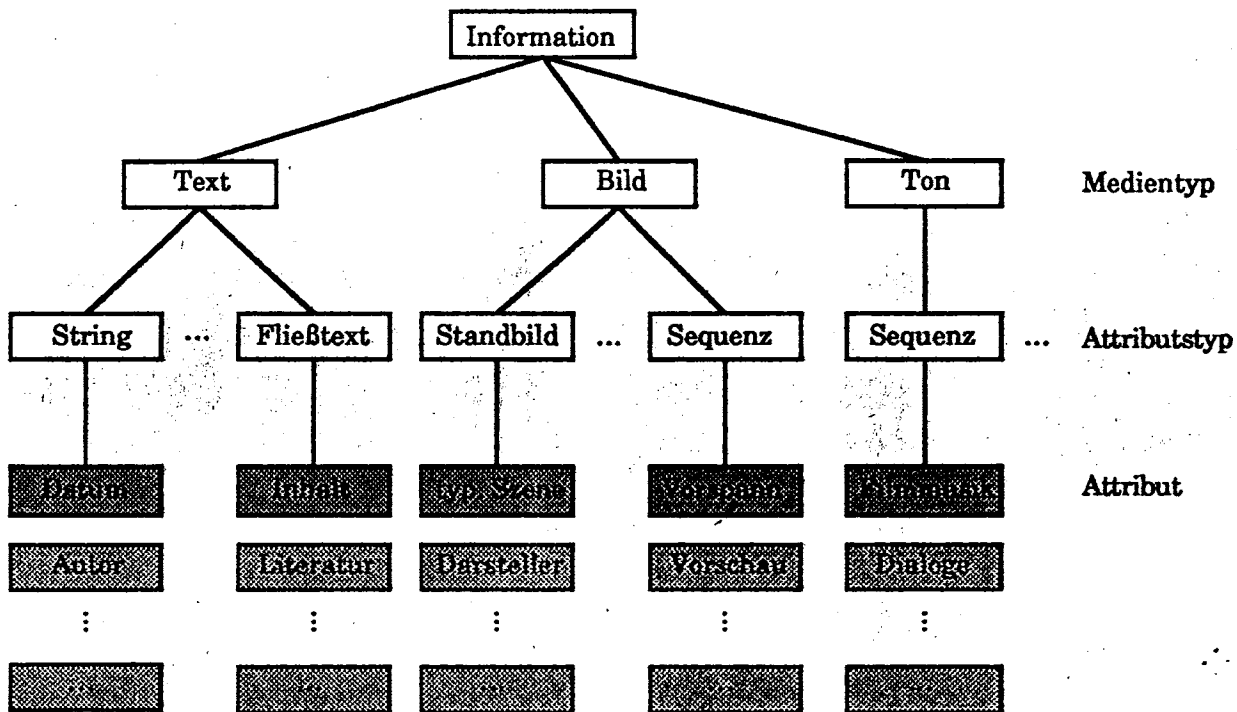


Abb.1: Informationstypen und Ausprägungen im Filmkatalog

Für die Verwaltung von Daten innerhalb eines Hyper-Dokumentes nennt [1] drei Möglichkeiten. Danach können die Daten in

- einer beliebigen, konventionellen, für andere Zwecke bereits existierenden (Netzwerk-/ hierarchischen/ relationalen/ objektorientierten) Datenbank
- einer speziell auf die Bedürfnisse eines Hypertext- Dokumentes zugeschnittenen Datenbank
- einem Hypertext-Dokument, ohne daß von außen eine separate Datenbank-Struktur zu erkennen ist

abgespeichert werden.

Der Entwurf des Filmkatalogs entspricht dem zweiten Ansatz. Jede Ausprägung eines Untertyps wird innerhalb einer typspezifischen Liste verwaltet. Als Beispiel aus Abb. 3 sei hier die Ausprägung "Autor" des Untertyps "String" genannt, die alle gespeicherten Drehbuchautoren enthält. Die Werte einer Ausprägung sind gekapselt und nur mit speziellen Methoden darstellbar und manipulierbar.

Die vorgesehene Verbindung der gespeicherten Informationen stellt ein wesentliches Merkmal für die Leistungsfähigkeit des Filmkatalogs dar. Zur verschiedenen Referenzierung von Objekten unterscheidet [1] zwischen hierarchischen Links, mit denen ein Dokument seine Hauptstruktur erhält und Querverweisen, die für das Aufzeigen von Querbeziehungen zwischen den Knoten des Hypertextes verwendet werden. In einer dritten Gruppe nennt er Annotationen und Anmerkungen, mit denen zusätzliche Knoten zur Erläuterung einzelner Konzepte an das ursprüngliche Hyper-Dokument angehängt werden können.

Alle Links innerhalb des Filmkatalogs werden erst zur Laufzeit dynamisch erzeugt. Nur so ist es möglich, die Forderung nach Erweiterbarkeit des Katalogs zu erfüllen. Grundoperationen eines auf dem Computer gestalteten Katalogs sind das Löschen, Einfügen und Verändern von Daten. Starre Links verursachen hohen Aufwand. Sie scheiden daher als praktische Realisierung aus. Die Strukturierung der gespeicherten Werte erfolgt mit Hilfe der durch das System vergebenen Filmnummer, mit der die Informationen verknüpft werden können.

Das System erlaubt ein beliebiges Navigieren über die dynamisch erzeugten Links innerhalb der Filmmenge. Der Suchraum wird von allen Suchbegriffen aufgespannt. Jede Attributliste stellt eine Dimension innerhalb dieses Raumes dar, z. B. Autor oder Filmtitel. Durch die Spezifikation von Anfragen werden die einzelnen Dimensionen des Hyperraums eingegrenzt. Weitere Anfragen können optional auf allen Daten oder einem bereits eingegrenzten Suchraum ausgeführt werden. Ergänzungen bzw. Umformulierungen von Anfragen erweitern den Suchraum bzw. grenzen diesen ein.

Attributswerte, die einer Anfrage entsprechen, werden aus den Listen ausgewählt und durch Verknüpfung die richtigen Filme ermittelt. Die sich ergebende Filmliste ist als Menge von Kanten im Hyperraum vorstellbar. Beim "Blättern" durch den Katalog werden nacheinander alle Knoten, die durch die Kanten verbunden sind, besucht.

Die Vermeidung starrer Strukturen und hier vor allem das Arbeiten mit den dynamisch erzeugten Links bilden die Grundlage für die Flexibilität des Katalogs. Die Bearbeitung der verschiedenen Attributstypen mit "gemeinsamen" Methoden vervollständigen diese Möglichkeiten. Die Erweiterung des Katalogs um neue Attributstypen ist möglich, um den Katalog veränderten Anforderungen anzupassen. Der Autor eines speziellen Katalogs kann Ausprägungen eines Attributstyps ergänzen, z.B. ein Attribut Hauptdarsteller. Das System stellt dann neue Karten, innerhalb der Datenbasis, mit den benötigten Methoden zur Verfügung. Das Bearbeiten des textuellen Datenbestands ist ohne Einschränkungen möglich. Entweder werden Ausprägungen einzelner Attribute verändert, entfernt oder hinzugefügt. Für sämtliche vorhandenen Informationen stehen Methoden zu deren Darstellung zur Verfügung. Die Verwendung dieser Methoden erfolgt unabhängig vom verarbeiteten Medientyp.

Realisierung

Ein modularer Softwareentwurf erfordert die Zerlegung des Systems in autonome Einheiten, die unabhängig voneinander spezifiziert, implementiert und getestet werden können. Der Entwurf des Filmkatalogs sieht die Zerlegung in die vier Module Lesermodul, Autorenmodul, Anfragebearbeitung und Datenmodul vor.

Für alle Benutzer mit Lesestatus wickelt das Lesermodul den kompletten Dialog zur Bedienung des Filmkatalogs ab. Bei der Erstellung des Moduls wurde auf eine einheitliche Erscheinungsform der Karten wertgelegt, die die Durchschaubarkeit des Systems für ungeübte Benutzer verbessert. Funktionale Eigenschaften des Moduls sind die Spezifizierung von Anfragen und das Blättern im gesamten bzw. durch eine Anfrage eingegrenzten Datenraum. Zur Realisierung dieser Leistungen wird auf das Anfrage- und das Datenmodul zurückgegriffen.

Das Autorenmodul ist nur für Benutzer mit Autorenstatus zugänglich. Es stellt über eine eigene Benutzungsschnittstelle Funktionen zum Pflegen und Erweitern des Datenbestands zur Verfügung. Der Datenbestand kann wahlweise durch Hinzufügen neuer Filme oder neuer Ausprägungen eines vorhandenen Attributstyps erweitert werden. Soll z.B. neben einer Inhaltsangabe und Literaturangaben eine weitere Ausprägung des Attributstyps Fließtext zur Aufnahme von Filmkritiken angelegt werden, ist diese Erweiterung des Datenbestands im Autorenmodul möglich. Darüber hinaus sind alle notwendigen Funktionen zur Instandhaltung der Daten vorhanden.

Zur Bearbeitung von Anfragen stellt das Anfragemodul die Schnittstelle zwischen dem Lesermodul und dem Datenmodul dar. Es übernimmt die Zerlegung komplexer Anfragen in Zugriffe auf interne Attributlisten des Datenmoduls. Alle Anfragen werden auf einem Stack gespeichert, so daß auf alte Anfragen zurückgegriffen werden kann. Das Ergebnis einer Anfrage wird dem Lesermodul in Form einer Liste von gefundenen Filmnummern bereitgestellt.

Das Datenmodul des Filmkatalogs übernimmt die Speicherung aller Filmkatalogdaten und stellt Methoden bereit, um auf den Datenbestand zuzugreifen. Jede Ausprägung eines Attributs ist mit der zugehörigen Filmnummer verbunden. Die Verwaltung dieser Relationen erfolgt in Listen. Die externe Sicht auf die verschiedenen Attributstypen ist durch die Bereitstellung gleicher Methoden für die Kunden des Moduls einheitlich. Aufrufende Module müssen daher nicht zwischen Text, Bild und Ton unterscheiden.

Abb. 4: Karte zur Anfrageneingabe.

Der Benutzer des Filmkatalogs kann eine Anfrage im Lesermodul formulieren. Dazu wählt er aus einer Liste aller Attribute einzelne aus und spezifiziert die gewünschten Wertebereiche. Die spezifizierten Attribute können wahlweise mit "und" bzw. "oder" verknüpft werden.

Beispiel:

Filmgruppe = "Arbeit und Hafen"
 und
 Produktionsjahr >"1980"

Diese komplexe Anfrage wird dem Anfragemodul übergeben und in zwei Anfragen zerlegt, die nacheinander an das Datenmodul weitergegeben werden. Für jede Anfrage stellt das Datenmodul eine Liste von gefundenen Filmen (Filmnummern) bereit:

1. Anfrage:	Filmgruppe = "Arbeit und Hafen"
Ergebnis:	{2, 5, 7, 27, 28, 39}
2. Anfrage:	Produktionsjahr >"1980"
Ergebnis:	{3, 7, 16, 25, 35, 39}

Die Ergebnislisten werden im Anfragemodul anschließend konjunktiv bzw. disjunktiv verknüpft:

Verknüpfung:	{2, 5, 7, 27, 28, 39} und {3, 7, 16, 25, 35, 39}
Ergebnis:	{7, 39}

Die Ergebnisliste wird dem Lesermodul zugänglich gemacht. Über die Filmnummern aktiviert das Lesermodul dann Methoden des Datenmoduls, zur Darstellung der Daten auf vorbereiteten Maskenkarten. Für den Anwender besteht die Möglichkeit, je nach Anfrageergebnis, die Anfrage zu erweitern oder einzugrenzen.

Fazit

Das Ziel des Arbeitsvorhabens war die Entwicklung eines besseren Filmkatalogs. Er sollte mehr und vielfältigere Informationen beinhalten, die leicht aktualisiert und erweitert werden können. Gleichzeitig sollten Funktionen angeboten werden, die den leichten Zugang zur gesuchten Information ermöglichen. Ideen zum Erreichen dieser Ziele wurden im vorliegenden System umgesetzt. Durch ein flexibles Typenkonzept ist die einfache Erweiterbarkeit um neue Informationsarten gewährleistet. Durch einfache aber trotzdem mächtige Anfragefunktionen ist es möglich, die Fülle von Information flexibel zu strukturieren und auf eine überschaubare Menge zu reduzieren.

Zwei Arten von Problemen sind im Laufe des Arbeitsvorhabens aufgetreten. Sie betreffen das gewählte Werkzeug HyperCard und konzeptionelle Schwächen des Entwurfs.

Die Probleme mit HyperCard begründen sich in der Diskrepanz zwischen der von HyperCard unterstützten und der gewählten Konzeption eines Hypertextes. Die übliche Vorgehensweise beim Speichern von Informationen in einem HyperCard-Stack, ist die feste Anordnung von Texten, Bildern usw. auf einzelnen Karten. Die Karten werden immer in dieser einmal festgelegten Form angezeigt. Ein flexibles Konzept erfordert jedoch Informationen erst während der Interaktion mit dem Benutzer anzuordnen. Sie wird zum Anzeigen aus der Datenbasis generiert und in vorgefertigte Maskenkarten eingetragen. Auch Listen mit Filmen werden direkt entsprechend den Wünschen des Benutzers gesucht, gruppiert und in Feldern auf bestimmten Karten angezeigt. Durch dieses sehr dynamische Verhalten unseres Stacks ist HyperCard überfordert. Dies macht sich insbesondere in Effizienzproblemen beim Anzeigen von Filmen bemerkbar.

Ein weiterer Punkt, der die Beschränkungen von HyperCard aufzeigt, ist die mangelnde Unterstützung einer modularen Entwicklung von Hyper-Dokumenten. HyperCard bietet keine Möglichkeiten, Schnittstellen für die einzelnen Module festzulegen oder Daten innerhalb der Module vor dem Zugriff durch andere Module zu schützen. Jedes HyperTalk-Programm kann auf jeden beliebigen Stack zugreifen und dessen Daten, dessen Struktur und sogar dessen Programme verändern. Eine getrennte Entwicklung einzelner Module durch verschiedene Personen wird erschwert, die Wahrscheinlichkeit von Fehlern durch Seiteneffekte erhöht. Das Arbeiten mit HyperCard hat aber auch gezeigt, daß die Entwicklung von Prototypen, durch das einfache Gestalten von Karten gefördert wird. Ein weiterer Vorteil von HyperCard ist die Offenheit des Systems. Nicht unterstützte oder zeitkritische Aktionen werden durch das Einbinden von externen Programmen ermöglicht.

Welche Grenzen hat das vorgestellte Konzept? Die Daten werden getrennt nach ihrem Typ in einzelnen Listen gespeichert. Die einzelnen Einträge werden über eine Filmnummer angesprochen. Dieses Vorgehen entspricht der Darstellung innerhalb eines Katalogs jede Information direkt einem Film zuzuordnen. Speicherung von Informationen, die einzelnen Attributswerten zugeordnet sind, sieht das Konzept zur Zeit nicht vor.

Die vorliegende Version des Filmkatalogs unterstützt die Suche nach Filmen über Begriffe, Namen, Themen usw. Die Suche nach Bildern wird bereits über Schlagwörter ermöglicht. Aus der Sicht des Nutzers sind Systeme wünschenswert, die den Umgang mit ungenau spezifizierten Anfragen gestatten, die alle Medientypen betreffen. Es fehlen Suchfunktionen in Bild und Ton. Kann die Suche nach einem Film über die Beschreibung markanter Filmszenen dann auch zum Klavierspieler im Casino und dem Mann im Riesenrad führen? Arbeiten zur Eingabe und Interpretation derartiger Anfragen sind ebenso wie Funktionen zur Suche in Bildern Gegenstand aktueller Forschung. Vorliegende Resultate machen wenig Hoffnung.

Literatur:

- [1] P.A. Gloor, Hypermedia Anwendungsentwicklung, Teubner Stuttgart, 1990.
- [2] W. Coy, "Film als Buch": Hyperdokumente zur Filmanalyse. in: P.A. Gloor, N.A. Streitz (Hrsg.) Hypertext und Hypermedia, Springer, 1990.
- [3] Forschungs- und Entwicklungsinstitut Film/Fernsehen an der Universität Bremen (Hrsg.), Film/Videokatalog'87, Bremen, 1987.

HYDRA

Filmanalyse im rechnergestützten Medienverbund

*Rita Abel, Jörg Blankemeyer, Wolfgang Coy, Jörg Geißler, Ferruh Mercan,
Ingo R. Meier, Oliver Schormann, Anja Tietjen, Torsten Rahn*

UNIVERSITÄT BREMEN INFORMATIK – PROJEKT HYPERCORPUS

Im Studiengang Informatik der Universität Bremen gibt es eine langjährige Tradition des Projektstudiums. Wesentliches Kennzeichen dieser Projekte, die sich über zwei Jahre erstrecken, ist es, einen Forschungsgegenstand unter verschiedensten Aspekten zu studieren und auf eine informationstechnische Gestaltung hin zu untersuchen. HYDRA ist ein studentisches Arbeitsvorhaben im laufenden Projekt HyperCorpus. Ziel von HYDRA ist die Entwicklung einer »Hypershell«, eines Programm- und Gerätepakets zur Unterstützung interaktiver Filmanalyse am Rechner. Damit soll erstmals die Möglichkeit geboten werden, ein rechnergestütztes »Filmbuch« [1, 3] zu schreiben, das neben der Analyse des Films und der kritischen Aufbereitung des Sekundärmaterials auch die unmittelbare Nutzung des Primärmaterials - nämlich des Films - zuläßt. Dies wird als praktischer Schritt zur Entfaltung des neuen Mediums Computer gesehen [4, 5]. Basis dieser Überlegungen ist die Nutzung des Bildschirmarbeitsplatzes zur Darstellung von Dokumenten unterschiedlichsten Charakters, also schriftlichen Texten, Bildern, Grafiken, Ton oder Film. Diese Forschungsansätze stehen derzeit unter dem Namen Hypertext oder Hypermedium in verstärktem Interesse, obwohl die Ansätze dazu bis in die Mitte der vierziger Jahre zurückverfolgt werden können. Trotz der offensichtlichen Beschränkungen der derzeitigen Technik kann man bereits jetzt mit verfügbaren Geräten einen Eindruck von der Umgestaltung einzelner Arbeitsprozesse durch den Einsatz solcher Technik gewinnen. Voraussetzung dazu ist es, einen geeigneten Arbeitsplatz zu analysieren, einen geeigneten Dokumentencorpus zu erstellen oder verfügbar zu machen und geeignete Techniken zur Hyperdokumentenbearbeitung konstruieren oder bereitstellen. HYDRA erfüllt diese Bedingungen.

Im Arbeitsvorhaben HYDRA wird ein einzelner Film in interaktiver Weise so präsentiert werden, daß vorhandene Text-, Grafik-, Standbild- und Toninformationen im Zusammenhang mit diesem Film präsentierbar gemacht werden. Aus technischen Beschränkungen heraus ist der Film selber nur in analog gespeicherter Form bereitgestellt (derzeit über einen ansteuerbaren S-VHS-Recorder, zukünftig soll ein ansteuerbarer Bildplattenspieler eingesetzt werden).

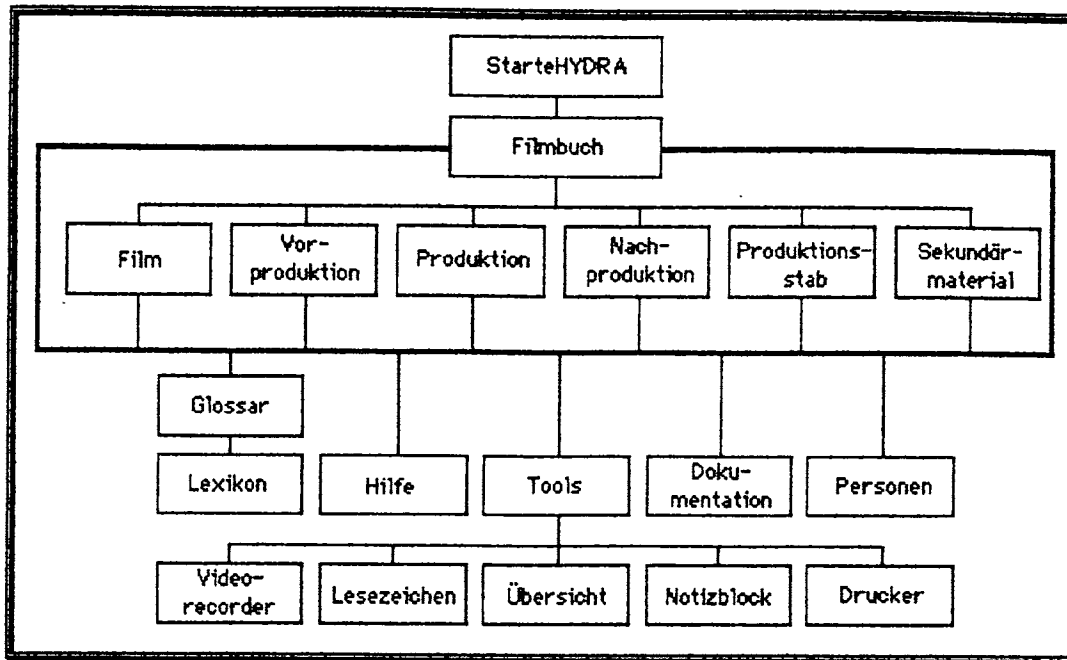


Abb.1 : Struktur der HYDRA-Shell

Die HYDRA-Shell unterstützt folgende *Basisfunktionen*:

- Film ansehen (Recorder ansteuern), auch teilweise nach Sequenzen oder Einstellungen;
- Analysematerial am Bildschirm lesen, blättern, nach Stichworten suchen;
- Lexikon nutzen;
- Referenzen nutzen;
- Notizen erstellen;
- Lesezeichen setzen;
- Online-Hilfefunktion.

LESEN UND DURCHSUCHEN (BROWSING)

Durch die Nutzung aller zur Verfügung gestellten Möglichkeiten der Programmierumgebung (derzeit HyperCard V2.0 und HyperTalk auf Apple Macintosh II), sind Benutzer der HYDRA-Shell in der Lage, von jeder Position aus jede andere zu erreichen.

Dies geschieht zum einen durch ein entsprechendes Pulldown-Menü, von dem aus zu den einzelnen Komponenten des Programms gesprungen werden kann, zum anderen durch eine Übersichtskarte (*map*), die die grobe netzartige Struktur der Shell zeigt (vgl. Abb.1), mit augenblicklichem Standort der Benutzer und den Möglichkeiten, diesen Standort zu ändern.

Darüberhinaus existieren innerhalb der einzelnen Komponenten Tasten (*buttons*), die mit der Maus angewählt werden können und die es zulassen, etwa Textmaterialien seitenweise zu lesen, Abschnitte zu überspringen, sich Inhaltsverzeichnisse anzeigen zu lassen oder bestimmte Textpassagen zu suchen (vgl. Abb. 4).

Durch die interne Protokollierung der Arbeit mit der Shell ist es schließlich möglich, bereits besuchte Stellen des Materials anzusteuern.

LEXIKON

Das *Lexikon* ist vom aktuellen Film unabhängig; es erklärt allgemein filmspezifische Begriffe mit Text, Grafik und/oder Animation. Ihm ist ein *Glossar* vorgelagert, welches Kurzbeschreibungen einzelner Begriffe bietet und für weitergehende Informationen in das Lexikon verzweigen läßt. Alle im Glossar und im Lexikon enthaltenen Begriffe sind der HYDRA-Shell zu jedem Zeitpunkt bekannt d.h. es werden automatisch nur solche Verbindungen (links) zwischen dem übrigen Material und dem Glossar/Lexikon hergestellt und verfolgt, die wirklich existieren.

REFERENZEN

Die HYDRA-Shell unterstützt sowohl von Texten ausgehende *Mehrfachreferenzen*, als auch globale Referenzierung eines Textes. Damit sind Autoren der Shell in der Lage, eine Passage, die mehrfach in einem Text auftaucht, nur einmal als Beginn einer Referenzliste definieren zu müssen. Von dieser Passage ausgehend, können nun beliebig viele betitelte Verbindungen zu anderen Medien (Text, Grafik, Audio, Video) angelegt werden. Leser werden durch deren Titel immer wissen, was sie erwartet, wenn sie einer bestimmten Verbindung folgen.

NOTIZFUNKTION

Die in der HYDRA-Shell integrierte *Notizfunktion* erlaubt es, zu jeder Zeit individuelle Anmerkungen zu speichern. Dabei kann es sich um private oder öffentliche Notizen handeln, die auch editiert und gespeichert werden können. Es werden sowohl Text-, Grafik- als auch Audionotizen (über Mikrofon und A/D-Wandler) unterstützt. Eine Online-Funktion erlaubt die Bewegung im Programm mit gleichzeitiger Änderung der Notizanzeige.

LESEZEICHEN

An jeder Stelle können im Programm *Lesezeichen* gesetzt werden, die direkte Sprünge zu bereits gesehenen Bereichen ebenso ermöglichen, wie das Anlegen von benutzereigenen Touren. Vorgegebene Sätze von Lesezeichen liegen der HYDRA-Shell bei, um die Materialfülle auf vorgegebenen Pfaden betrachten zu können.

HILFE

Das *Hilfemodul* bietet im wesentlichen drei Möglichkeiten: die Hilfestellung zur aktuellen Position im Programm (audiovisuell), die Zuschaltung einer Online-Hilfe, die zu jeder geänderten Position automatisch Benutzungshinweise liefert, sowie die Erläuterungen zu speziellen Bereichen des Programms (Paletten, Menüs, etc.).

Mit den Benutzerfunktionen wird in der HYDRA-Shell auf das Primärdokument Film und dessen Ansteuerprogramme zugegriffen. Ebenso wird der Zugriff auf die Materialien zum Film, zu Personen unterstützt. Das Modul FILM dient der inhaltlichen Erschließung eines elektronisch gespeichert vorliegenden Filmes.

Es unterteilt sich in folgende fünf Abschnitte:

- Sequenzen-/Take-Verwaltung
- Markenverwaltung
- Videorecordersteuerung
- Indexregister
- Autoreferenz.

Die Sequenzen/Take-Verwaltung hat hierbei die Aufgabe, im Dialog mit den Benutzern einen Film in die filmisch-strukturellen Einheiten Sequenz und Take zu unterteilen. Jede Einheit wird textuell beschrieben und dessen Filmabschnitt durch Start- und End-Timecode festgelegt. Für die Festlegung des Filmabschnittes ist die Markenverwaltung verantwortlich. In Zusammenarbeit mit der Videorecordersteuerung (bei Einsatz des Filmmediums Video) wird ein Film abschnittsweise bearbeitet und die bereinigten Ergebnisse der Sequenz/Take-Verwaltung übergeben. Ein derart bearbeiteter Film kann über die strukturell-optische Anspielung hinaus auch begrifflich erfaßt werden. Dies ermöglicht das filmabhängige Indexregister. In diesem werden alphabetisch sortiert Begriffe zu einem Film von den Benutzer zusammengefaßt. Diese verweisen dann auf bestimmte einspielbare Filmabschnitte. Über die Möglichkeit hinaus, einen Film inhaltlich und begrifflich zu beschreiben, versetzt die Autoreferenz die Benutzer in die Lage, eine ausgewählte Textstelle automatisch mit einem Filmabschnitt zu verknüpfen. Hierbei fungiert das Indexregister als Lieferant für Verknüpfungsbegriffe.

Im PERSONEN-Modul werden sämtliche Personen der Filmanalyse aufgeführt, d.h. auch Schauspieler, Kamera, Regie etc. Es ist auf verschiedene Arten möglich, von anderen Stellen des Programms, in den Personenstapel zu gelangen. Eine Möglichkeit ist, durch ein Inhaltsverzeichnis, eine bestimmte Person nach dem Namen auszuwählen. Eine andere Möglichkeit ist der automatische Aufruf einer Person an einer anderen Stelle der HYDRA-Shell, indem z.B. der Name an der Position ausgewählt wird. Es ist auch eine Filterfunktion geplant, die ermöglicht, durch Bestimmen von Eigenschaften, die die Personen erfüllen müssen, eine Personen-Gruppe zur Auswahl zu extrahieren.

Zu jeder Person sind umfangreiche Daten gespeichert (mit Bild, sofern ein Bild vorhanden ist). Da einige Daten schon im Programm vorkommen, werden auch Referenzen auf andere Stellen herangezogen, wie etwa auf das Sekundärmaterial.

Auf das SEKUNDÄRMATERIAL in der HYDRA-Shell kann auf zwei verschiedene Arten zugegriffen werden. (In beiden Fällen mittels einer eigenen interaktiven Menüsteuerung) Zum einen ist es nach seiner Art strukturiert, so daß es über einen Menüpunkt Sekundärmaterial und dann nach Kriterien wie »Kritiken«, »Interviews«, »Produktionsberichte« und »Vorlage« erschließbar ist (vgl. Abb. 2). Zum anderen ist das Material auch nach Inhalten des Produktionsablaufes gegliedert - und zwar in die Bereiche »Vorproduktion« (mit solchen Aspekten wie Regie, Produzent, Idee, Vorlage u.a.), »Produktion« (Kamera, Licht, Ton u.a.) und »Nachproduktion« (Schnitt, Musik u.a.). Die drei Bereiche des Produktionsablaufes beinhalten also ein chronologisch geordnetes »Filmbuch« (vgl. Abb. 3). Für diese beiden verschiedenen Herangehensweisen über Sekundärmaterial und Produktionsablauf wurde das Material entsprechend vorstrukturiert, so daß es nur einmal im Programm gespeichert werden muß. In beiden Fällen werden aber auch Informationen und Daten aus anderen Bereichen des Programms herangezogen und aufbereitet.

Umgekehrt sorgen von uns erstellte sogenannte HyperTalk-Handler dafür, daß das Material (bzw. bestimmte Auszüge daraus) in anderen Modulen der HYDRA-Shell zur Verfügung steht. Konkret bedeutet das, daß bei einem Aufruf des Sekundärmaterials in einem anderen Modul (z.B. im Personenstapel wünscht jemand Kritiken zu einer bestimmten Person) das System zum Sekundärmaterialstapel springt, dort alle (zum Wunsch passenden) verfügbaren Materialstellen aufammelt und anschließend interaktiv zur Anzeige bringt. (Soll heißen: jede weitere Stelle wird erst auf Knopfdruck angesprungen und es ist jederzeit möglich, diesen Sekundärmaterialaufruf abzubrechen und zum aufrufenden Modul zurückzukehren.)

Die beiden folgende Grafiken sollen die Struktur des Sekundärmaterials wiedergeben.

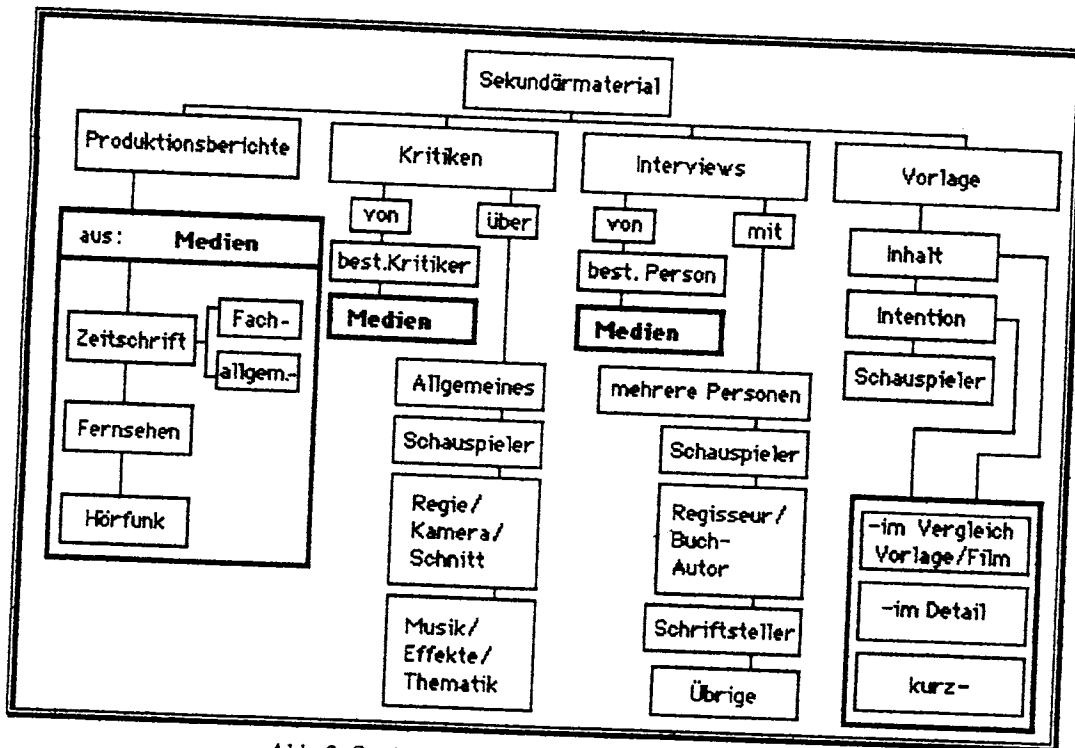


Abb. 2: Struktur des Sekundärmaterials - nach Art

Die HYDRA-Shell besteht aus einer Menge von HyperCard Stapeln, die zur Speicherung von Daten und zur Verwaltung der Funktionen dienen. Sie läuft unter HyperCard 2.0 auf Apple Macintosh mit 12" oder 13" Monitor. Ein zweiter Monitor/Fernsehbildschirm dient der Wiedergabe des Films vom Videorecorder. Zur Wiedergabe des Films haben wir uns für einen Panasonic S-VHS Video-Recorder entschieden. Dieser ist mit einer RS-232C Schnittstelle ausgestattet, die die externe Steuerung des Video-Recorders vom Macintosh ermöglicht. Dazu braucht die HYDRA-Shell entsprechende externe Kommandos (XCMDs). Zum bildgenauen Aufsuchen einzelner Szenen wird ein externer Timecode-Generator eingesetzt (der Ziellauf des Videorecorders ist für den Einsatz in HYDRA zu ungenau). Der Timecode-Generator wird an den Macintosh und an den Videorecorder angeschlossen und von der HYDRA-Shell ausgelesen. Längere Zugriffszeiten auf einzelne Bildpassagen sind mit dem Recorder unvermeidlich. Eine bessere Lösung wäre der Einsatz eines Bildplattenspielers, wobei die Aufnahmekapazität der Platte eine neue Beschränkung darstellt.

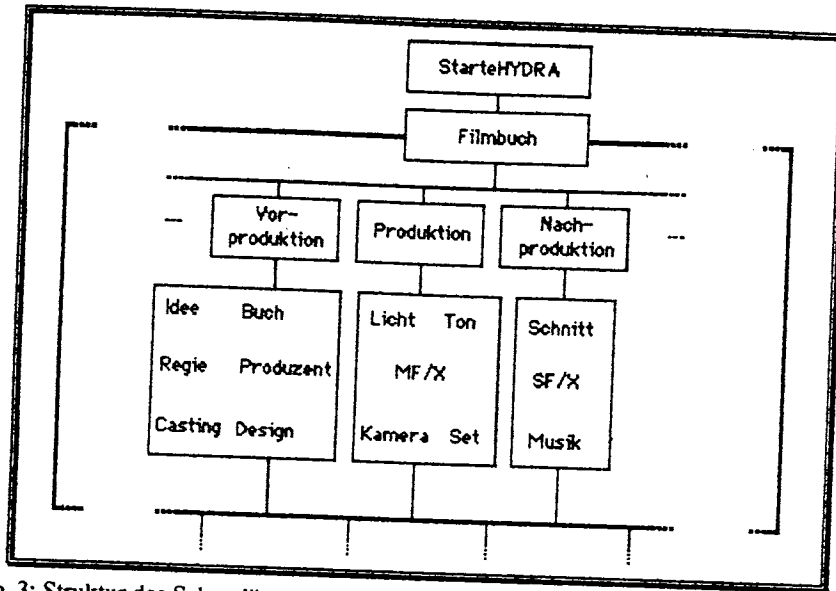


Abb. 3: Struktur des Sekundärmaterials - nach Inhalten (im Kontext des Gesamtsystems)

BEISPIELSITZUNG MIT DER HYDRA-SHELL

Nach Auswahl des Filmbuchs "Shining" wurde zum Produktionsstab verzweigt. Hier wurde der Begriff KAMERA angewählt und auf der entsprechenden Karte befand sich das Stichwort STEADICAM.

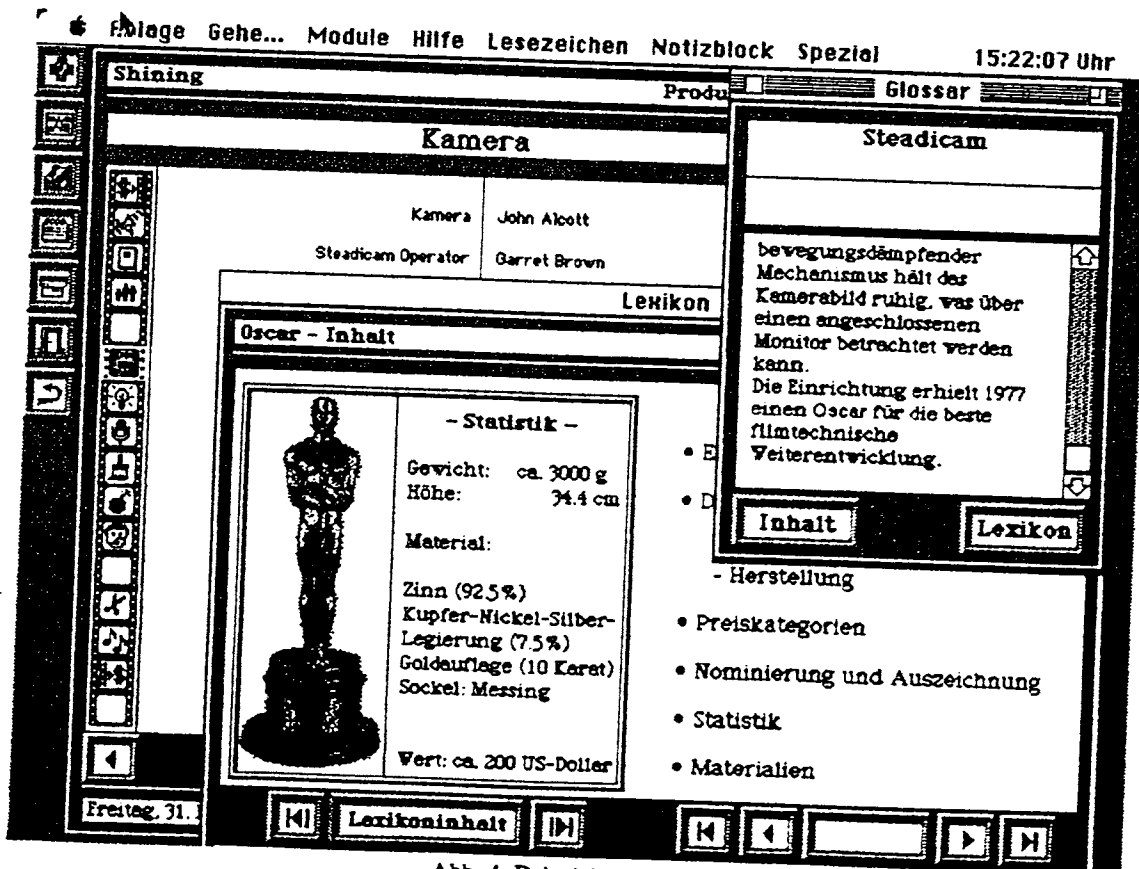


Abb. 4: Beispielsitzung 1

Dieses konnte angewählt werden, worauf die Shell zum Glossar verzweigte. Von dort aus kam man über den Begriff "Oscar" zum Lexikon.

BEISPIELSITZUNG EINES AUTORS MIT DER HYDRA-SHELL:

An einem Text der NEWSWEEK über den Film "Shining" mußten Veränderungen vorgenommen werden. Rechts ist die Autoren-Palette zu sehen. Nach diesem Arbeitsschritt soll zu einem anderen Modul verzweigt werden.

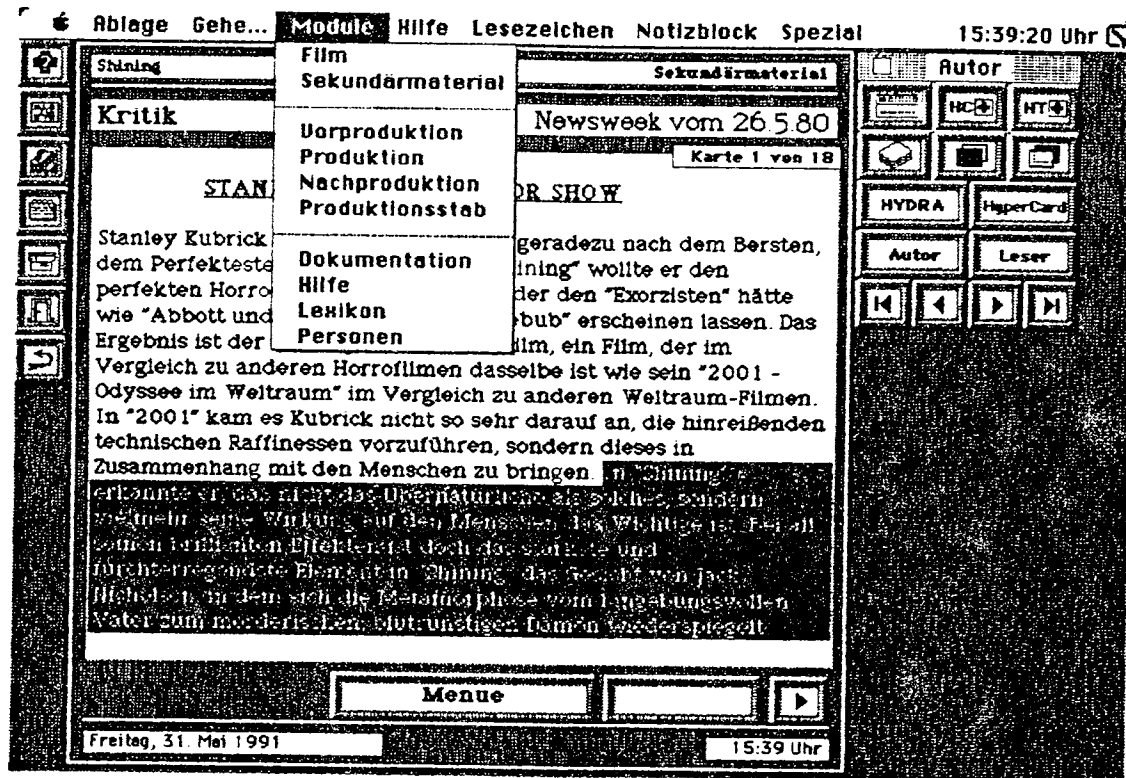


Abb. 5: Beispielsitzung 2

LITERATUR

- [1] Coy, W.: 'Film als Buch': Hyperdokumente zur Filmanalyse, in P.G. & N. Streitz (Hrsg.): Hypertexte/Hypermedia. Berlin-Heidelberg-New York-Tokio: Springer 1990
- [2] Deleuze, G.: Das Bewegungs-Bild — Kino 1. (Übers. von U. Christians & U. Bokelmann) Frankfurt/M: Suhrkamp 1989,
- [3] Godard, J.-L.: Einführung in die wahre Geschichte des Kinos. (Übers. von F. Grafe & E. Patalas) München: Hanser 1981,
- [4] Negroponte, N.: 'Books Without Pages', in Proc. IEEE International Conference on Communications IV, 1979
- [5] Nelson, T.H.: Computer Lib — Dream Machines. Redmond, Wash.: Microsoft Press 1987, Überarbeitete Neuauflage der Ausgabe von 1974, Chicago: Hugos Book Shop
- [6] Virilio, P.: Krieg und Kino — Logistik der Wahrnehmung. (Übers. von F. Grafe & E. Patalas) Edition Akzente München: Hanser 1986, 191 p.

Literatur und Kunst - Vorbereitungen f. eine Ästhetik der Hypersysteme bei R.Roussel, H.Bellmer u. O.Wiener

Was mit den Hypersystemen möglich zu werden scheint - die tendenziell uneingeschränkte Kombination, Vermischung, Überblendung etc. der einzelnen Medien - ist als Zielprojektion in der Literatur u. bildenden Kunst spätestens mit Beginn des 20.Jh.s präsent. In der Auseinandersetzung mit den neuen Medien Fotografie/Film/Funk erscheinen in den traditionellen Medien Formen, Verfahren, Objekte und Zusammenstellungen, die unter den Titeln Collage, Montage, Aleatorik etc. Texte, Bilder, Alltagsfundstücke, letztlich beliebige Materialien kombinieren. In Futurismus, Dada, Surrealismus werden die tradierten Gattungsgrenzen in Frage gestellt, aufgebrochen, wenn auch nicht aufgelöst: die "Rahmen" bleiben erhalten, die Bild- u. Buchform, die Orte (Galerie, Museum, Bibliothek etc.), wenn auch ganz andere Dinge in ihnen auftauchen (eine Werbeanzeige im lyrischen Text, eine verrostete Schraube auf einer Leinwand), oder eine Ausstellung auf die Straße verlegt wird. Erst mit den weiteren technischen Entwicklungen besonders im elektronischen Bereich, d.h. der Transformierung aller Medien in digitalisierte Informationssysteme wird auch die tatsächliche Auflösung der Systemgrenzen möglich.

Dennoch gibt es im literarischen und künstlerischen Vorfeld so etwas wie radikale Versuche, tradierte Formen zu sprengen, und zwar gerade *innerhalb* einer Gattung. Als Beispiel können die Texte R.Roussels, die Konstruktionen (Plastik, Zeichnung, Fotografie) H.Bellmers u. die zwei "Romane" O.Wiener gelten:

- Roussels eigenartiges Konstruktionsprinzip, eine Geschichte, ein Drama, eine Szene als logische Kette zwischen zufälligen, per Assoziation, Homonymie, Alliteration gefundenen Elementen zu erfinden, wobei die "Körper" als durch Zeichenfolgen in ihren Bewegungen definierte Maschinen erscheinen; reflektiert nicht bloß - autopoietisch - den literarischen Text "corpus". Das Verfahren löst klassische Subjekt- und zugehörige Geschichtsprojektionen wirklich auf - eine "anonyme" literarische Produktion, wie sie immer erst der "Computerpoesie", der elektronischen (durch Hypertext ausweiteten) Kombinatorik zugeschrieben wird.

- Bellmers sogenannte "Puppe", d.h. die über Jahrzehnte betriebene Konstruktion einer anthropomorphen Gestalt, ergänzt durch Zeichnungen, nochmals transformiert durch Fotografien, experimentiert mit einem in der bisherigen Kunstgeschichte unbekanntem Verhältnis von (innerer/äußerer) Wahrnehmung - materiellem Substrat (Körper) - Zeichen. Unter den Begriffen Balance, Amalgam, Anagramm versucht Bellmer die ästhetische Artikulation als beständigen Austauschprozeß von Energien.

"Schmerzen", und technischen Ersetzungen darzustellen - eine bewegliche Verknüpfung der Medien, die doch nicht *beliebig* passiert.

- Wieners bisherige zwei größere Texte, "Die Verbesserung von Mitteleuropa" und "Nicht schon wieder..." beschreiben - "radikaler Konstruktivismus"? - Welt und Subjekt, Helden und Gegenspieler, Städte und Landschaften, als *Einbildungen* elektronischer Systeme, *Namen* für wiederkehrende Datenverknüpfungen, notwendige Projektionen von Netzen, die ihren Informationsfluß erhalten müssen - das ist wohl nicht nur als literarische Satire beiseite zu legen.

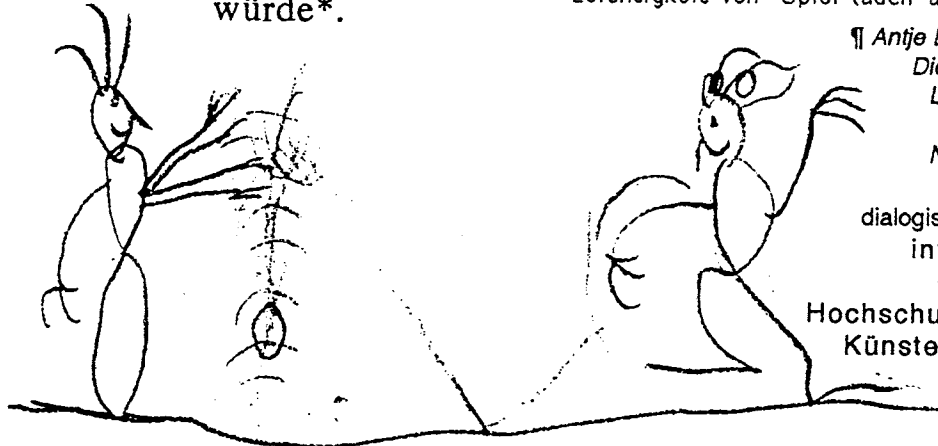
Dialogische Künstlerrolle

(von Handlochkart' zu hai'pæ:ka:d)

von Kurd Alsleben, Hamburg und Detlev Fischer, Coventry

Zur nötigen Beschleunigung
unseres Brief-Dialogs,
lieber Detlev, beabsichtige
ich, Dich folgend gleich
mehrfach "anzuspielen"*,
wie Volker Lettkemann sagen
würde*.

¶ 'Sich die Bälle zuspieren' bedeutet m.E.,
gemeinschaftlich etwas durchsetzen zu wollen.
'Anspielen' (Zeichnung: Jonas) setzt auch
sympathisches Verhalten voraus, aber es besteht
keineswegs ein Gemeinschaftsprojekt. Der schöne
Ausdruck "an-spiele'n" bedeutet verstecktes
hinweisen, anfangen, thematisieren und die
Leichtigkeit von Spiel (auch 'anzettelnder Eros').



¶ Antje Eske, Werner Justen,
Dieter Kaitinnes, Volker
Lettkemann, Emanuel
Müller und Nicola
Nissen sind Mitglieder
einer Arbeitsgruppe
dialogischer Datenkunst der
interdisziplinären
Computerei der
Hochschule für bildende
Künste Hamburg, deren
Ergebnisse hier mit
zugrundeliegen.

Lieber Kurd,

-- ein Gegenstück zum Anspielen:
das sich Anbieten. *

Dann aber nimmt sie den Ball, wie er kommt, und
versucht, zu verwandeln.

-- Im Dialog steht es mir einigermaßen frei, auf
welchen Teil oder Zug der Äußerung der Anderen ich
antworte.

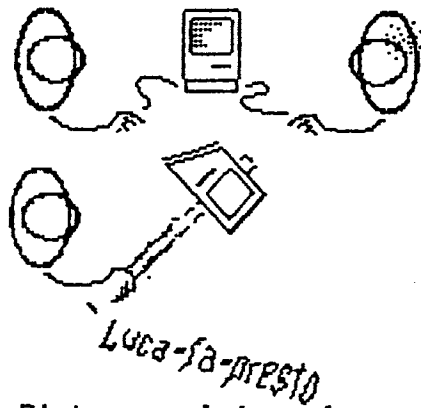
--Vielleicht am elegantesten geht es zu, wenn ein
gerichtetes Anspiel (ein Thematisieren, indirektes
Auffordern) in der Szene des Dialogs wegschmilzt: ich fühle
es nicht als ein Hin und Her (und will dann auch nichts mehr
durchsetzen). Einhergeht ein lässiges, ruhiges Körpergefühl (wie
alt das Wort wohl erst ist)

Im anderen Sinne (als: auf etwas anspielen) ist das Wort reicher,
ein Stück Welt, mehr als operierbarer Ball selbst an uns
operierend, oder ein subject ist in doppelte
Aufmerksamkeit gerückt, wenn die andere mitspielt.

Aber: wir sind doch froh, wenn wir einmal gegen die
Spielregeln mit Gegenspielern spielen und kein
(argument) machen (und glühen dann froh beim Eintritt
in die (fremde) Atmosphäre).

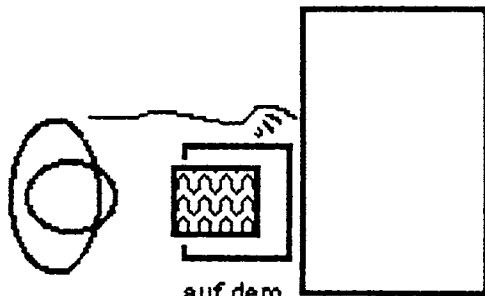
* (Fußball: niemand bietet sich
an; also versucht er's alleine,
bis er sich verdrückt und den
Ball verliert)

Wenn bloß das Reden von
Atmosphäre nichts
Schwabbeliges hätte.
Ob vor dem Fond Natur oder
in Brief- oder Zeiträumen: die
gute Stimmung ist der
Glückslohn des Dialogs.



Die Pictura zeigt *unten* mit dem auf Ersparen, Zeitökonomie zielenden Motto den Barockmaler Luca Giordano und andererseits *oben* uns, die wir uns Zeit nehmen mögten!! - schnell ist ja das Werkzeug, echtzeitschnell.

Hinter dem Computer -als Medium, Formulierungsmedium - tritt der Andere, Nitmensch auf.



1960 erlebte Künstlerrolle: fremde Äusserung begehrend

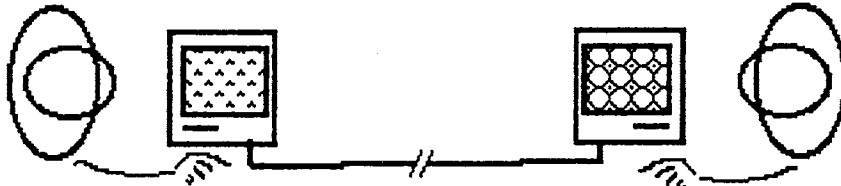
auf dem Plotter, Zeichnung als erwartete Äusserung

Computer (in interaktiver Benutzung) mit vermuteter innerer Zustandswahrnehmung

Bei DESY habe ich mit Cord Passow 1960 künstlerische Zeichnungen mittels Computer gemacht. Die Beziehung zwischen dem Computer und uns hatte interaktive, dialoghafte Gestalt. Ich vermutete im Computer generalisierende interne Zustandswahrnehmungen, die er mitteilen könnte. Computer und Plotter waren weit davon entfernt, ein Werkzeug zu sein. Ich empfand den Computer als fremdes anderweites Wesen, er deutete mir den anderen Menschen an. Damals befand ich mich in einer Künstlerrolle, in der ich die Computerzeichnungen

zwar hervorrief, ich sie aber staunend auf dem Plotter als Ausdruck des

Computers sah. Ich erlebte eine Künstlerrolle, in der ich mich nicht als Autor vorfand, sondern fremde Äusserung begehrte. Der virtuelle Andere war sichtlich nicht mein Agent oder Assistent - er war Dialogist, ein Mitmensch. In der Entwicklung bis heute ist der andere Mensch auch hinzugekommen, durch Telematik oder vermittels der Autorenprogramme. Das Rechen-Werkzeug wurde Medium.



Hinzugekommene dialogische Künstlerrolle

Telematik. Ferne oder Simulation sind für den Dialog keine vorrangigen Momente

Dialogische Rollen des Anderen

Aus den Teilen des Dialogs, dessen ortloser Schauplatz der Computer ist, lese ich die andere wieder auf, setze sie als eingehörte Stimme und als eingebildetes Gesicht wieder zusammen.

--tote Teile, deren Stil ihnen beiläufig ein wenig Leben einsoffliert. Deshalb ist die Form der Äußerung so wichtig, die typographische Gestalt ist eine Stimm-hilfe für mich Innerlich-Laut-Leser

Ich halluziniere ja nicht nur die Stimme der anderen, sondern - sofern ich die Äußerung als an mich gerichtet empfinde - eine Gemeinsamkeit, die erinnerten wirklichen Gemeinsamkeiten mit ihren para-stimm-sprachlichen Merkmalen (sich-einigen-auf-eine-{Ironie}-Stimmung), Gesichte, Feldlinien des Ungesagten) nachgebildet scheint. Wie schön wäre es, an Gedankenübertragung zu glauben. Die Maschine wird hierbei unsichtbar.

Paul sagte anschließend - oder war es vorher? - : "I consider it very indecent to talk to an audience any longer than a minute." (Ich finde es sehr unfein, irgend länger als eine Minute zu einer Zuhörerschaft zu sprechen)

Paul Wolf, Hamburg, arbeitet zur Zeit an einer neuen Zeichensetzung. Wir dürfen u. a. ein Antwortzeichen und ein eine systematische Anzeige von Gedanken-ebenen erwarten.

Wenn das Schirmbild wieder dazwischenschlägt, und ich das Buchstäbliche lese, bin ich im Zustand der Kritik und möchte jedes Wort zerreißen und bloßstellen. Alle meine Kraft habe ich daran. Ich kann das aber nur im Dienste eines Herren oder einer Herrin tun.

Mai 1991. Niederschmetternd ist es zu sehen, wie Germanistikstudenten in Hamburg (in einem Canetti-Seminar) - nicht gegeneinander, sondern, wie durch saubere Pissoirtrennwende geschieden, strikt nebeneinander, eine bevorstehende Belohnung aus der Gedankenwelt halluzinierend (der Lektor scheint als Person unwichtig, wird eher geschnitten als angespielt) übers Thema reden. Ein Seminarraum mit 60 Studenten scheint das sicherste Mittel, die schöne Weichlichkeit eines Dialogs auszuschließen.

Literatur

(¶: Kurd Alsleben, Δ: Detlev Fischer)

- ¶Δ Alsleben, Kurd: *Ästhetische Redundanz*. Quickborn 1962.
- ¶Δ ders.: *Antwortnot und Spiel*. Manifest. Hamburg 1986.
- ¶ ders.: *In welcher Form soll man (Autor) über Diskette formulieren?* In: G. Lobin (Hrsg.): Paderborner Arbeitspapiere. Universität Paderborn 1987.
- ¶Δ ders.: *Diskettentypographie*. In: Palavecino, J.C.: *Cibernetica, Ordinatores i Teoria de Sistemas*. Universität Barcelona 1987. Und: Hypertext-Diskette. Griffelkunstvereinigung, Hamburg 1988.
- ¶ Alsleben, K., M. Lehnhardt et al.: *Der ästhetische Begriff der Beteiligung*. FUniv. Hagen 1977.
- ¶Δ Alsleben, K. und Antje Eske (Hrsg.): *KükoCokü*. HbK Hamburg 1989ff
- ¶ Decker, E. und P. Weibel: *Vom Verschwinden der Ferne. Telekommunikation und Kunst*. Köln 1990
- Δ Deleuze, Gilles und C. Parnet: *Dialogues*. London 1987.
- ¶Δ Gerstner, Karl: *Die Anfänge der Partizipationskunst*. In: Alsleben, K. und A. Eske (Hrsg.): *KükoCokü*. Hamburg 1989.
- ¶ Gesellschaft für Mathematik und Datenverarbeitung (Hrsg.): *MMK '90*. Königswinter 1990.
- ¶ Glaser, Peter: *Das Kolumbus-Gefühl*. In: Wieckmann, J. (Hrsg.): *Das Chaos Computer Buch*. Reinbek 1988.
- ¶ Finke, Harald: *Subdialog*. Hamburg 1987
- ¶Δ Fischer, Detlev: *Dialoghafte Figuren*. HbK Hamburg 1990.
- ¶Δ ders.: *Basic Notes about Dialogue*. Coventry Polytechnic 1991.
- ¶ Herzog, J. und M. Carl: *Lexikon Neue Medien*. Berlin (DDR) 1989
- ¶Δ Höpel, Ingrid: *Emblem und Sinnbild. Vom Kunstbuch zum Erbauungsbuch*. Frankfurt a.M. 1987.
- Δ Kierkegaard, Søren: *Concluding Unscientific Postscript* (1846). Princeton 1968.
- Δ Kittler, Friedrich A.: *Aufschreibesysteme 1800 - 1900*. München 1987.
- ¶Δ ders.: *Synergie von Mensch und Maschine*. In: *Kunstforum* 98. 1989.
- ¶ Lehnhardt, M. und T. Ammann: *Die Hacker sind unter uns*. München 1985.
- ¶ Lehnhardt, M., G. Meißner et al.: *Hacker für Moskau*. Reinbek 1989.
- ¶ Lingner, Michael: *Strategien ästhetischen Handelns*. In: *Kunst+Unterricht* 152/1991.
- ¶ Lipp, Achim: *Kunst im Netzwerk*. Kunsthalle Hamburg 1987
- Δ Longman, Karl: *Flexible Arrangement in Overcoded Media*. Coventry Polytechnic 1991.
- Δ Lohenstein, Daniel Casper: *Sophonisbe* (1680). Reinbek 1968.
- ¶ Maaß, Susanne: *Mensch - Rechner - Kommunikation. Herkunft und Chancen eines neuen Paradigmas*. Universität Hamburg 1984
- ¶ Malsy, Victor: *El Lissitzky*. Mainz 1990.
- Δ Moravec, H.: *Mind Children. The Future of Robot and Human Intelligence*. Cambridge 1988.
- ¶ Müller, Roland: *Vor uns das Kommunikations-Paradies? High-Tech und Selbstverwirklichung*. In: *Schweizer Monatshefte*, Januar 1990.
- ¶ *Nake*, Frieder, für Dich einen freundlichen Gruss.
- Δ Ong, Walter S.J.: *Interfaces of the World*. Ithaca, London 1977.
- Δ Penrose, Roger: *The Emperor's New Mind*. New York 1990.
- ¶ Reichlin, Urs: *Der Spielende in der Rolle des Spielers und des Spielverderbers*. Universität Zürich 1976.
- ¶ Schuller, Marianne: *Dialogisches Schreiben*. In: Hahn, B. und U. Isselstein (Hrsg.): *Rahel Levin Varnhagen*. Göttingen 1987.
- ¶ Sikora, H. und F. X. Steinparz: *Computer und Kommunikation*. München, Wien 1988.
- Δ Virtanen, Reino: *Conversations on Dialogue*. University of Nbraska 1977.
- Δ Waite Group: *Tricks of the HyperTalk Masters*. Indianapolis 1989.
- ¶ Walther, F. E. und M. Lingner: *Zwischen Kern und Mantel*. Klagenfurt 1985
- ¶Δ Wieckmann, J., M. Lehnhardt et al.: *Das Chaos Computer Buch. Hacking made in Germany*. Reinbek 1988.
- ¶ Zemanek, H. und P. Goldschneider: *Computer. Werkzeug der Information*. Berlin, New York 1971.

Integration von Expertensystem- und Hypertext-Techniken am Beispiel des Systems WISKREDAS

Fahri Yetim

Universität Konstanz
FG Informationswissenschaft
Postfach 5560, D-7750 Konstanz

Abstract

In diesem Beitrag werden die Aspekte der Integration von Expertensystem- und Hypertext-Techniken am Beispiel des Konstanzer wissensbasierten Systems WISKREDAS, welches einen komplexen Entscheidungsvorgang in einer Verwaltungsumgebung durch kooperative Wissensverarbeitung informationell unterstützt, vorgestellt. Durch den hier unternommenen Integrationsversuch, der sowohl die Wissensrepräsentation als auch die Dialogkomponente betrifft, wird eine Verbesserung der Systemleistung erhofft, die vor allem auch in der Kooperation mit dem Benutzer und in der Unterstützung intuitiver Prozesse des Benutzers besteht.

1. Einleitung

Expertensysteme, die u.a. mit dem Ziel angetreten sind, Entscheidungshandlungen zu automatisieren oder zumindest weitgehend zu unterstützen, haben jedoch ihr ursprüngliches Hauptziel, Expertenleistungen auf dem Computer verfügbar zu machen, nur in sehr eng begrenzten Anwendungsgebieten erreichen können. In schwach-strukturierten Problembereichen, wo das relevante Wissen nicht vollständig erfaßt und im System repräsentiert werden kann, wird vielmehr auch die Einbeziehung der menschlichen Erfahrung notwendig. Es wird daher häufig der Anspruch an Informationssysteme gestellt, daß sie nicht nur die in einer Organisation anfallenden, vorhandenen oder benötigten Daten bzw. das Wissen effizient und effektiv verwalten, sondern auch den Benutzer dabei unterstützen, in einer konkreten Problemsituation die notwendige Information zur Problemlösung zu erarbeiten.

Hypertexte, die als eine integrierte Form von Datenbanktechnologie, Wissensmanagement und multimedialer Präsentationstechnik (Text, Ton, Graphik, bewegte Bilder) angesehen werden, können durch ihre nicht-linearen Eigenschaften und multimedialen Möglichkeiten einer immer wieder geförderten *flexiblen Darstellung von Wissen* und einer ebenso *flexiblen Erarbeitung von entscheidungsrelevanter Information* entgegenkommen (Kuhlen 1991). Die Integration der dem Hypertext-Konzept inhärente Eigenschaften in ein Expertensystem bedeutet eine Anreicherung der Systemeigenschaften und somit auch die Überschreitung der funktionalen Grenzen des Systems (vgl. Yetim/Dambon 1991), wodurch die Effizienz und Effektivität der Entscheidungsfindung auf schwach-strukturierten Problemfeldern erheblich verbessert werden können. Die durch

die funktionalen Erweiterungen erreichte Systemleistung geht über die automatisch durchgeführten Berechnungen hinaus, besteht aber nicht nur in der textuellen und graphischen Aufbereitung von Informationen, sondern auch in der Möglichkeit des freien Navigierens in der Systemwissensbasis. Ferner bietet sich für den Benutzer die Möglichkeit, eigene Lösungsvorschläge, Positionen, Argumente usw., wie bei IBIS-artigen Systemen (vgl. Conklin 1987), einzubringen.

In Verlaufe dieses Beitrags ist die Vorstellung solcher im Rahmen des Systemprototypen WISKREDAS bereits durchgeführten Integrationsversuche intendiert. Im folgenden Kapitel wird zunächst die Systemarchitektur von WISKREDAS kurz beschrieben. Anschließend erfolgt eine Diskussion der Notwendigkeit der Erweiterung des Systems um die hypertextuellen Eigenschaften. Im weiteren Kapitel werden die Integrationsaspekte im einzelnen ausführlich behandelt.

2. Projektbeschreibung

Das System WISKREDAS (wissensbasiertes Kreditabsicherungssystem) wurde im Rahmen des an der Universität Konstanz angesiedelten Sonderforschungsbereichs 221¹, "Verwaltung im Wandel", entwickelt (vgl. Dambon et al. 1989). Das System dient der Unterstützung eines Kreditsachbearbeiters bei der Beurteilung von Bürgschaftsanträgen und entstand nach eingehenden empirischen Untersuchungen in einer halbstaatlichen deutschen Bank. Die Unterstützung besteht im wesentlichen darin, daß das System das Wissen, das zur Bearbeitung eines Kreditantrags eines Existenzgründers benötigt wird, verwaltet, zum Teil automatisch einholt, dieses nach Regeln, die auf dem Fachwissen des Sachbearbeiters beruhen, verarbeitet und schließlich, wenn die informationelle Basis als ausreichend eingeschätzt wird, einen Entscheidungsvorschlag macht, der im Ergebnis eindeutig sein muß: Empfehlung auf Ablehnung oder Übernahme der beantragten Bürgschaft.

WISKREDAS verfügt über fünf parallel und selbständig arbeitende Komponenten (vgl. Abb. 1), die sich gegenseitig über ein Mailsystem Messages schicken und deren Funktion nachfolgend kurz erläutert wird: Der 'Case Base Manager' (vgl. Dambon 1988) verwaltet die Fälle in einem Fallarchiv; dort sind alle abgeschlossenen sowie die in Bearbeitung befindlichen Fälle abgespeichert. Im 'Entscheider' (vgl. Wolf 1988) werden die vorhandenen Daten aufgrund bestehender Regeln ausgewertet und bei ausreichender Informationsgrundlage eine Entscheidung berechnet und vorgeschlagen. Bei noch fehlenden Informationen stellt der Entscheider Anfragen per Mailsystem an den 'Wissens-

¹ Das Projekt wurde unter der Leitung von Prof. Dr. Rainer Kuhlen (Lehrstuhl für Informationswissenschaft an der Universität) und der Mitarbeit von Peter Dambon, Fabian Glasen, Martin Thost, Michael Wolf und schließlich des Autors (F.Y.) entwickelt. Die weiterführenden Arbeiten hinsichtlich hypertextueller Erweiterungen werden vom Autor im Rahmen seines Dissertationsvorhabens vorgenommen. An dieser Stelle sei allen Mitarbeitern für ihre Zusammenarbeit und Vorleistungen gedankt.

basierten Ressourcenmanager' (vgl. Glasen 1990), der Zugang zu internen und externen Wissensressourcen (z.B. Online-Datenbanken) hat. Er transformiert u.a. die Anfragen des Entscheiders in die jeweilige Hostsprache und integriert rückkommende Antworten zu einer Antwort auf die vom Entscheider gestellte Frage. Der 'Informationsbewerter' (vgl. Thost 1990) bewertet abschließend die vom Ressourcenmanager erarbeiteten Informationen hinsichtlich der Glaubwürdigkeit der Informationslieferanten und versucht, widersprüchliche Informationen zu homogenisieren. Die Dialogkomponente, die mit allen anderen Komponenten kommunizieren kann, übernimmt den Dialog mit dem Benutzer. Sie enthält eine Teilkomponente zur graphischen hypertextartigen Präsentation und Eingabe von Fällen (Dambon/Yetim 1990) und eine Teilkomponente zur Erklärung der Systemleistungen (Yetim 1991). Diese beiden Teilkomponenten werden in Kapitel 4. mit Bezug auf ihre hypertextuellen Eigenschaften vorgestellt.

Aufgrund der Charakteristika des untersuchten Problemfeldes, d.h. aufgrund der Verschiedenartigkeit der Fälle, des unendlichen Wissensraumes, aus dem neue Anträge auftauchen können, aber auch aufgrund von juristischen und Akzeptanzbedenken wurde kein automatisches System, sondern ein kooperatives System konzipiert. Wichtiger als die vom System erstellten Berechnungen ist somit dessen Funktion hinsichtlich der *Informationsaufbereitung für die intuitive Einschätzung* des Benutzers. Im folgenden wird diskutiert, warum zum Erreichen dieser Funktionalität die Integration von Hypertext-Techniken notwendig ist.

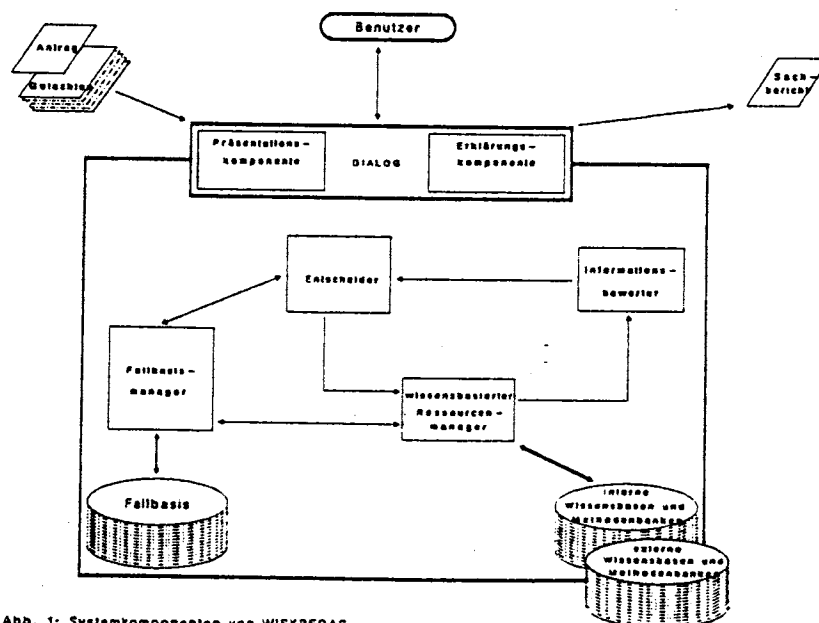


Abb. 1: Systemkomponenten von WISKREDAS

3. Zur Notwendigkeit der Hypertextmethodologie für WISKREDAS

Die Entscheidung zur Ablehnung bzw. Bewilligung eines Antrags basiert auf der Einschätzung der Erfolgsaussichten der zu gründenden Firma, konkret auf der Prognose, ob dem Antragsteller nach Abzug aller Ausgaben und Verpflichtungen genügend Mittel für den Lebensunterhalt verbleiben. Die Entscheidung stützt sich nicht nur auf wirtschaftliche Berechnungen, sondern auch auf Erfahrungen des Sachbearbeiters und auf die Gutachten von verschiedenen externen Experten (z.B. von Kammern, Verbänden usw.), die zu den wichtigsten Projektparametern Stellung nehmen. Alle diese Aspekte und unterschiedliche Informationstypen müssen berücksichtigt werden und in einer Entscheidungssituation in den Entscheidungsprozeß einfließen. Hier kommt daher sowohl der Wissensrepräsentation als auch der Präsentation eine große Bedeutung zu, deren Realisierung durch die dem Hypertext-Konzept inhärente Flexibilität wie folgt unterstützt wird:

(1) Die in den externen Gutachten enthaltenen Expertenmeinungen und Aussagen, die in Form von nicht-formalisierten Texten vorliegen, müssen analysiert und verwaltet werden. Die Hilfestellung des Hypertextes besteht zum einen in der flexiblen Organisation und der kreativen Verknüpfung dieser Texte, zum andern in der Erkennung der bereits bestehenden Zusammenhänge. Hierzu werden die Expertenmeinungen als Hypertext-Knoten in der Wissensbasis abgelegt und durch semantische oder argumentative Bezüge strukturiert, die vom Benutzer auf der Basis individueller Interpretation dieser Knoteninhalte manuell gesetzt und somit quasi zu systeminternen Meinungen umgeformt werden. Während der Nutzung können diese Informationen nach unterschiedlichen Prinzipien strukturiert und dem Benutzer zur intuitiven Einschätzung bereitgestellt werden.

(2) Wie bereits erwähnt, ist das System WISKREDAS in der Lage, selbst zu berechnen, ob die vorhandenen Informationen eine Entscheidung zulassen oder nicht. Wichtiger als diese vom System erstellte Berechnung ist jedoch seine Funktion hinsichtlich der Informationsaufbereitung für die intuitive Einschätzung des Benutzers, die die vorhandenen Informationen in einer Form verfügbar machen, in der sie in Entscheidungsprozessen effizient und effektiv wirksam werden können. Hypertexte bieten eine verbesserte Mensch-Maschine-Kommunikation durch Anwendung neuer informationstechnischer, multimedialer Möglichkeiten. Durch die Integration der Hypertext-Techniken können Daten und Wissen als Informationen (also selektiert und zweckgerichtet) in geeigneter Form präsentiert und dem Sachbearbeiter mächtige und komfortable Werkzeuge zur Verfügung gestellt werden, mit denen er im Wissensbestand des Systems frei und gezielt navigieren kann. Zur Darstellung dieser Informationen bieten sich unterschiedliche Präsentationstechniken an: mehrere inhaltlich verwandte Hypertext-Knoten lassen sich gleichzeitig darstellen. Unterschiedliche Wissenstypen können in Entscheidungssituationen dem Sachbearbeiter zur intuitiven Einschätzung bzw. zur

Abwägung der Argumente verschiedener Experten bereitgestellt werden.

(3) Zu jedem Antrag muß ein Sachbericht erstellt werden, worin die Entscheidung begründet wird. Entscheidungen müssen also nachvollziehbar und erklärbar sein. Das System muß über eine Erklärungskomponente verfügen, die in der Leistung über die bisherigen Trace-Funktionen weit hinausgeht. Der häufig aus der Systementwicklung geäußerte Hinweis, daß alle Entscheidungen des Systems durch Zurückverfolgung der logischen Schlüsse nachvollzogen werden, hat kaum praktische Relevanz, da ab einem schon recht niedrigem Komplexitätsgrad dies zu aufwendig und faktisch nicht mehr möglich ist (vgl. Kuhlen 1989). Durch die Hypertextkomponente, die als Dialogwerkzeug dient, wird der Benutzer des Systems aktiver in den Prozeß der Entscheidungssuche einbezogen (vgl. Yetim 1991, Yetim/Dambon 1991). Die auf dem Hypertext-Konzept beruhende Erklärungskomponente ermöglicht, die Vorgehensweise bzw. das Verhalten des Systems in einer flexiblen und akzeptablen Weise kritisch zu überprüfen. Durch die Navigationsmöglichkeit ist der Benutzer im Erklärungsdialog nicht mehr an einen einzigen vorgegebenen Weg gebunden (wie in herkömmlichen Trace-Funktionen, wo man eine vorgegebene Kette von Gründen bzw. Argumenten zu verfolgen hat), sondern er kann sich selbst Erklärungen erarbeiten.

Aufgaben, die WISKREDAS mit seinen Schlußfolgerungsmethoden nicht oder nur ungenügend beherrscht, werden durch die Integration der Hypertext-Techniken auf den Benutzer verlagert, wobei hier seine Intuition und Erfahrung durch die Flexibilisierung der Aufbereitung bzw. Präsentation der Informationen zuverlässig unterstützt wird.

4. Die Integrationsaspekte

Zum Erreichen der oben beschriebenen Funktionalität, d.h. zum Zusammenführen beider Konzeptwelten (Hypertext und WISKREDAS) bedarf es zum einen der Erweiterung der zugrundeliegenden Wissensrepräsentationsstrukturen, zum anderen einer objekt-orientierten Sicht auf die Wissensbasis des Systems, um dadurch flexible Dialoge zu ermöglichen. In den folgenden Unterkapiteln wird auf diese beiden Aspekte ausführlicher eingegangen.

4.1 Wissensbasis-Konzepte von WISKREDAS: Die objekt-orientierte Sicht

Die grundlegende Repräsentationseinheit von WISKREDAS ist der sogenannte Makroframe, in dem das gesamte statische Fallwissen in einer Framestruktur modelliert ist (vgl. Dambon 1988). Der Makroframe ist der Prototyp, das Skelett eines jeden Einzelfalles, aufgrund dessen neue Fälle sowohl generiert als auch bearbeitet werden können. Den Makroframe kann man sich als Netzwerk vorstellen, dessen Knoten Konzepte (Frames) und dessen Verknüpfungen Beziehungen zwischen den Konzepten repräsentieren. Jeder

Frame besteht wieder aus einer Menge von Eigenschaften (Slots) und zu jeder dieser Eigenschaften existieren sowohl Vorschriften, welche zulässige Einträge (Constraints, Defaults, Ranges, bedingte Prozeduraufrufe etc.) spezifizieren, als auch statische Texte (z.B. Definitionen, Fehlermeldungen etc.). Um die Wissensbasis mit hypertextuellen Elementen anzureichern, wurde der Makro-Frame durch die Erweiterung der einzelnen Frame-Konzepte um entsprechende Slots ausgebaut (vgl. Dambon/Yetim 1990).

Aus dem Makroframe, in dem das Fallwissen konzeptuell beschrieben ist, werden Einzelfälle generiert. Diese stellen konkrete Ausprägungen (Instanzen) des Makroframes, d.h. sie enthalten nur konkrete, fallspezifische Slot-Einträge (Daten). Darüberhinaus wird vom System jeweils nach einer Fallbearbeitung einen 'Entscheidungsbaum' generiert, der in die Entscheidung (Berechnung) eingehende Informationen enthält.

Die erst im nachhinein konzipierte Dialog-Komponente, die auf dem Hypertext-Konzept basiert und bislang unter den Komponenten des Systems am wenigsten weit entwickelt ist, erfordert eine objekt-orientierte Sicht auf die Elemente der Wissensbasis des Systems. Entsprechend müssen alle Elemente der Wissensbasis aus der Sicht dieser Komponente als Objekte oder besser, um bei der Hypertextterminologie zu bleiben, als Knoten betrachtet werden.

In WISKREDAS werden zwei Typen von Knoten unterschieden: **Basis-Knoten**, die die Basis-Konzepte der Wissensbasis darstellen, i.e Regeln, Fakten, Texte etc., und (dynamisch) **zusammengesetzte Knoten**, die mehrere Basis-Knoten enthalten, welche zueinander in einem gewissen thematischen Zusammenhang stehen². Beispielsweise stellt der vom System während der Problemlösung generierte dynamische Entscheidungsbaum einen solchen zusammengesetzten Knoten dar, dessen Inhalt die Form eines UND/ODER-Baumes aufweist. Weitere zusammengesetzte, inhaltlich kohärente Knoten, die während des Dialogs zu Präsentations- und Erklärungszwecken dynamisch kreiert werden, werden später in Kapitel 4.2. und 4.3. in Zusammenhang mit der Präsentations- und Erklärungskomponente behandelt. Im folgenden werden zunächst die Basis-Knoten und die Verknüpfungen zwischen ihnen betrachtet:

Basis-Knoten:

* **Faktum-Knoten:** der Faktum-Knoten bezeichnet ein Faktum in der Wissensbasis und stellt den gemeinsamen Oberbegriff für die Konzept-, Slot- und Instanz-Knoten dar. Diese Bezeichnung dient in erster Linie zur Unterscheidung zwischen atomaren Ausdrücken und Regeln.

- **Konzept-Knoten** repräsentieren die Konzepte des Makro-Frames und sind in der Prolog-Notation wie folgt realisiert, wobei jeder Konzept-Knoten einen Konzeptnamen und eine Liste von Slots enthält.

² wie die sog. 'composite nodes' (vgl. Halasz 1988, Garg/Scacchi 1988).


```
concept(concept_name, [slot1,..., slotn]).
```

- **Slot-Knoten** repräsentieren die einzelnen Aspekte der Konzepte. Da zu einem Slot mehrere Facetten existieren können, kann es mehrere Knoten desselben Slots geben, mit jeweils unterschiedlichen Inhalten. Bei den unteren Beispielen geht es um die Beschreibungen unterschiedlicher Aspekte des Slots '*amount*' (Gesamtsumme) des Konzeptes '*turnover*' (Umsatz). Der Unterschied liegt in ihren Facetten, die, wenn der entsprechende Knoten angewählt wird, jeweils unterschiedliche Aktionen ausführen oder Informationen geben. Beispielsweise wird im ersten Slot-Knoten angegeben, in welchem Intervall (*within_ranges*) der Eintrag liegen darf, während der zweite Angaben darüber enthält, auf welches Konzept (hier '*cash-flow*') die Einträge des Slot '*amount*' des Konzeptes '*turnover*' Auswirkungen haben.

```
slot(turnover,amount,dictate(within_range(10000,10000000))).
```

```
slot(turnover,amount,if_added(affect('cash-flow'))).
```

- **Instanz-Knoten** repräsentieren die Instanzen der Konzepte, d.h. die konkreten Ausprägungen. Die folgende Prolog-Notation eines Instanz-Knotens enthält Angaben über die Fallnummer (*11*), den Konzeptnamen (*turnover*), den Slot-Namen (*amount*) und schließlich den konkreten Slot-Eintrag, der aus einem Tupel besteht und unterschiedliche Angaben enthält, deren weitere Erläuterung an dieser Stelle jedoch irrelevant ist.

```
instance(11,turnover,amount,(actual,900801,chamber_a,0.65,300000)).
```

- * **Regel-Knoten** repräsentieren die Regeln in der Wissensbasis. Ein Regel-Knoten besteht - in Analogie zu einer Regel - aus einem Knoten für den Regelkopf und einem für den Regelrumpf. In den folgenden Beispielen in Prolog-Notation besteht der erste Regel-Knoten aus dem Regelkopf '*a*' und dem Regelrumpf '*a1,a2;a3...an*', wobei die miteinander durch UND/ODER verknüpften Elemente des Regelrumpfes wiederum jeweils entweder einen Faktum-Knoten (wie '*a1*') oder den Kopf einer anderen Regel (wie '*a2*') darstellen. Im letzteren Fall liegt eine Verkettung mehrerer Regel-Knoten vor.

$$a :- a_1, a_2; a_3 \dots a_n.$$

$$a_1.$$

$$a_2 :- a_{21}, a_{22} \dots a_{2m}.$$

- * **Grafik-Knoten** enthalten Graphiken;
- * **Text-Knoten** beinhalten Texte (vgl. Abb. 2). Die Inhalte der Text-Knoten und Grafik-Knoten können nicht vom System interpretiert werden. Ihre Semantik wird durch vordefinierte Hypertext-Verknüpfungen, die sowohl vom Benutzer als auch vom System gesetzt werden können, explizit und somit auch für das System interpretierbar gemacht. Text-Knoten werden wiederum bezüglich ihrer Inhalte unterschieden in:
 - **Definitions-Knoten**, welche Definitionen bzw. Begriffserklärungen enthalten;
 - **Argument-Knoten**, welche Experten-Meinungen bzw. Aussagen enthalten und
 - **Rechtfertigungs-Knoten**, die textuelle Beschreibungen für die Rechtfertigung einer Regel bzw. des kausalen Zusammenhangs, der in einer Regel formuliert wurde, und für die Bedingungen ihrer Anwendung enthalten (wobei in den textuellen Beschreibungen selbst wiederum Verzweigungsmöglichkeiten bestehen). Diese Knoten erlauben dem Benutzer, unabhängig von der Strategie der Regelbearbeitung des Systems (z.B. in Vorwärts oder Rückwärtsverkettungsmodus), seine eigene Strategie bei der Suche zu wählen. Durch die unterschiedlichen Abstraktionen der Regel und die dazugehörigen textuellen Beschreibungen mit ihrer jeweiligen Form der Abstraktion bietet sich die Möglichkeit zur flexiblen Handhabung von Erklärungstexten.

Basis-Verknüpfungen:

Zwischen diesen als Basis-Knoten behandelten Wissensstrukturen existieren verschiedene Relationen, die als Hypertext-Verknüpfungen betrachtet werden und u.a. auch zur Darstellung des Wissens auf unterschiedlichen Detaillierungsebenen und aus unterschiedlichen Perspektiven dienen.

- * **Intra- und interkonzeptuelle Relationen**, wie *is-Slot*, *is-a* und *instance-of*, relationieren unterschiedliche Konzepte bzw. Aspekte der Faktum-Knoten.
- * **Abstraktionsrelationen** sind z.B. *refineable* und *is_parameter_of*. Die Relation *refineable* besteht dann zwischen zwei Frame-Konzepten (die hier als Konzept-Knoten betrachtet werden), wenn ein Slot eines Frames selbst wieder ein Frame (d.h.

ein Konzept-Knoten) ist, der seinerseits eine ausführliche Beschreibung enthält. Die Relation *is_parameter_of* besteht zwischen zwei Konzept-Knoten dann, wenn ein Konzept Berechnungsparameter für das andere Konzept ist.

- * **Argumentationsrelationen**, wie *pro*, *contra* und *neutral*, die den Inhalt von Expertenmeinungen in den Argument-Knoten klassifizieren (vgl. auch Abb. 2). Diese Verknüpfungen können auf zwei Arten gesetzt werden: (a) durch den Benutzer, indem er die als Argument-Knoten organisierte Expertenmeinung zu einem bestimmten Konzept(-Knoten) inhaltlich interpretiert und die entsprechende Argumentationsverknüpfung zwischen diesen beiden Knoten setzt; (b) durch das System: dazu werden die von der Systemkomponente 'Informationsbewerter' gelieferten Bewertungen von Expertenmeinungen in Abhängigkeit von Vergleichsgrößen (z.B. als Schwellenwerte) interpretiert und die passende Verknüpfung automatisch hergestellt. Ein konkreter Sachverhalt, wie etwa eine bestimmte Verknüpfung zwischen zwei Knoten, wird in einem Instanz-Knoten repräsentiert:

instance(11,turnover,pro,(actual,900801,technologie_center,0.85,document_11_32)).

Der Inhalt dieses Instanz-Knotens ist wie folgt zu interpretieren: Zum Konzept 'turnover' des aktuell zu bearbeitenden Falles (Fall-Nr: 11) existiert im Dokument 'document_11_32' eine Stellungnahme vom Technologiezentrum (*technologie_center*) mit einem Bewertungsfaktor '0.85'. Diese Stellungnahme stellt ein Pro-Argument bezüglich des Konzeptes 'turnover' dar. Weitere Angaben beziehen sich auf die Aktualität (*actual*) und das Eingangsdatum des Dokumentes (01.08.90).

4.2. Die Präsentationskomponente

Da die Systemleistung von WISKREDAS u.a. auch in der *Informationsaufbereitung für die intuitive Einschätzung* des Benutzers besteht, kommt der Präsentationskomponente eine besondere Bedeutung zu. Beim Stand der Entwicklung kann die Präsentationskomponente, die den Dialog mit dem Benutzer und die Präsentation von Informationen auf dem Bildschirm übernimmt, folgende Funktionen wahrnehmen:

a) graphische Vernetzung der frame-basierten Konzepte des Diskursbereichs: die Darstellung des Netzes kann unter unterschiedlichen Gesichtspunkten geschehen, wie beispielsweise in Abb.2 unter Berücksichtigung der Berechnungen. D.h. die Verknüpfungen zwischen den Konzepten geben an, wie die Konzepte in der Berechnung aufeinander wirken. Im Wissensnetz kann navigiert und durch das Anwählen eines dargestellten Konzeptes (wie z.B. 'turnover') weitere Informationen über das Konzept

präsentiert werden;

b) tabellarische Darstellung der Konzepte mit ihren fallspezifischen Daten (Einträge); damit können alle Eigenschaften und Eigenschaftsausprägungen eines gewählten Konzeptes sowie Spezifikationsvorschriften, Erklärungen und ggf. Fehlermeldungen in verschiedenen Windows bereitgestellt werden.

c) Präsentation von Argumenten aus Gutachten zu bestimmten Konzepten.

d) Rekonstruktion einer Fallhistorie, aus der nicht nur die zeitliche Entwicklung bestimmter Einträge, sondern auch die dazu gehörenden Informationslieferanten und die vom System vorgenommene Bewertung hervorgehen.

e) gleichzeitige Präsentation von verschiedenen Fällen in verschiedenen Windows.

f) Darstellung der Arbeitsweise und Kommunikation zwischen den einzelnen Komponenten.

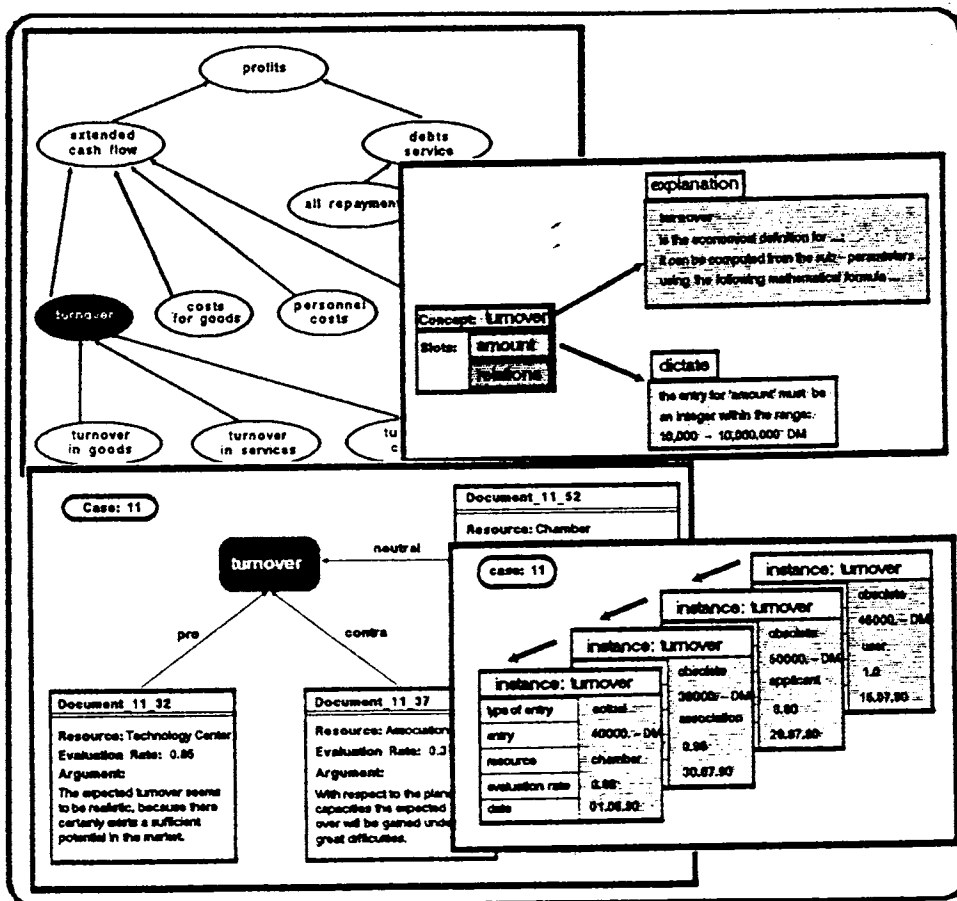


Abb. 2.: Präsentation der Informationen in WISKREDAS

4.3. Die Erklärungskomponente

In WISKREDAS wurde zur Entwicklung einer Erklärungskomponente auf das Hypertext-Konzept zurückgegriffen. Bisherige Arbeiten zu dieser Komponente, die in Kapitel 4.3.1 kurz vorgestellt werden, beschäftigten sich mit der systemseitigen Bereitstellung von situationsadäquaten Ausdrucksmöglichkeiten (Fragen), durch die der Erklärungsbedarf vom Benutzer geäußert wird. Im weiteren Kapitel werden u.a. auch mit Hilfe eines Beispiel-Dialogs der aktuelle Stand der Überlegungen darüber vorgestellt, wie die Erklärungsinhalte bestimmt, strukturiert und präsentiert werden können.

4.3.1. Benutzerfragen für Erklärungsdialoge

Es gibt eine Reihe von Fragen, die für die Erklärungskomponente relevant sind. Die für die aktuelle Situation relevanten Fragen lassen sich aus der Menge der möglichen Fragen unter Berücksichtigung der pragmatischen Aspekte, d.h. der Erklärungssituation, der Dialoghistorie und der Benutzerklasse durch das System automatisch bestimmen und dem Benutzer zur Auswahl anbieten (vgl. Yetim 1991). Dazu wurden Fragen als Objekte aufgefaßt und deren semantische und pragmatische Aspekte mittels Frames repräsentiert³ (vgl. Abb 3.).

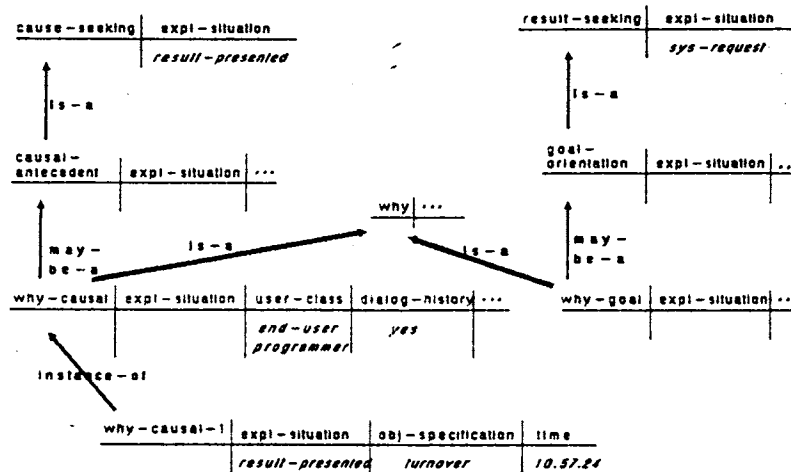


Abb. 3.: Eine Beispiel-Repräsentation

Bei den Fragen wird zwischen den Frageklassen und den konkreten Fragen unterschieden. Sowohl die Frageklassen als auch die konkreten Fragen werden auf unterschiedlichen Abstraktionsebenen über die Spezialisierungsbeziehung 'is-a' modelliert, wie z.B. die Frage-Klassen 'cause-seeking' und 'causal-antecedent', sowie die konkreten Fragen 'Why' und 'Why-causal'. Darüberhinaus bestehen zwischen den konkreten Fragen und den zugehörigen Frageklassen die Rollenbeziehung 'may-be-a', durch die die konkrete

³ Die Fragen können im Hypertext-Kontext auch als getypte Hypertext-Verknüpfungen (vgl. Yetim 1989) aufgefaßt werden.

Frage die Eigenschaften der Frageklasse erbt. In Abb. 3 ist beispielsweise der oberste Frame *'cause-seeking'* nur für die Erklärungssituation *'result-presented'* (d.h. wenn das System ein Ergebnis präsentiert hat) relevant, wohingegen der Frame *'result-seeking'* in der Erklärungssituation *'sys-request'* (d.h. wenn das System Fragen gestellt hat) zum Tragen kommt. Zur Anfragezeit vererbt sich die Erklärungssituation auf alle Unterklassen. Aufgrund der bestehenden *'May-be-a'*- Beziehung sowohl zwischen der Frage *'Why-goal'* und der Klasse *'goal-orientation'* als auch zwischen der Frage *'Why-causal'* und der Klasse *'causal-antecedent'* wird es möglich, in der aktuellen Situation zur Anfragezeit durch die Vererbung der Erklärungssituation zu bestimmen, welche Variante der 'Warum'-Frage relevant ist. Die Darstellung der Fragen auf unterschiedlichen Abstraktionsebenen erlaubt, die Fragen entsprechend der Benutzerklasse zu qualifizieren: z.B. kann sich unter Berücksichtigung der aktuellen Benutzerklasse anstatt der gültigen Variante (z.B. *'Why-goal'*) deren Abstraktion (*'Why'*) als Menu-Option qualifizieren.

Nach dem der Benutzer im Frage-Menü eine Frage selektiert hat, wird ein Instanz-Frame generiert, der beschreibt, auf welches konkrete Objekt sich die Frage in gegebener Situation bezieht. In unserem Beispiel handelt es sich um das Konzept 'Umsatz' (*turnover*). Der Instanz-Frame wird in die Dialog-Historie aufgenommen.

4.3.2. Kontextbildung durch frage-orientierte Operationen in der Wissensbasis⁴

In einer hypertext-basierten Erklärungsumgebung führen die Benutzerfragen dazu, daß nach semantischen und pragmatischen Gesichtspunkten aus den strukturell möglichen Antworten eine Teilmenge ausgesondert und dadurch der Bereich, der Kontext⁵, bestimmt wird, in dem der Benutzer selbst die für ihn sinnvollen Antworten auf eine Frage erarbeiten kann. Ähnlich wie die Komposition mehrerer inhaltlich verwandter Hypertextknoten zu einem Knoten (sog. *'composite node'* wie in Garg/Scacchi 1988) können für einen Fragetyp relevante Basis-Knoten zusammen mit ihren ebenso für die Frage relevanten Verknüpfungen zu einem einzigen Knoten (sog. *'Kontext-Knoten'*) zusammengefaßt werden. In Abhängigkeit vom Fragetyp lassen sich unterschiedliche Kontext-Knoten bilden, wie z.B. *'Warum-Knoten'*, *'Was-Knoten'* sowie *'Wie-Knoten'* für die Kontexte von Warum-, Was- und Wie-Fragen.

Die Form der Erklärung (bzw. die Strukturierung des erklärungsrelevanten Wissens) ist somit vom Fragetyp abhängig. Während z.B. *'Was'*-Fragen eher deskriptive Antworten verlangen (z.B. Definitionen, vorformulierte, explizite Erklärungen auf Fragen in der Form *'Was bedeutet x?'*), können sich *'Wie'*-Fragen beispielsweise darauf beziehen, auf welche

⁴ Die Überlegungen befinden sich noch in der Implementationsphase

⁵ Mittels einer Kontextbildung kann auch die Benutzerkonfusion und der Orientierungsverlust in einer Hypertextumgebung minimiert werden.

Art und Weise das System zu seinem Schluß kam. Die Form der Erklärung auf 'Warum'-Fragen (bzw. 'Warum-nicht'-Fragen) kann dagegen argumentativ sein, wobei auch Mischformen möglich sind. Im Dialog kann sowohl in einem Kontext als auch zwischen unterschiedlichen Kontexten navigiert werden, wodurch das Rezipieren und Aufsuchen der die eigentliche Erklärung konstituierenden Einheiten durch den Erklärungssuchenden nach seinen Bedürfnissen entsprechend erfolgt.

In Abb. 4. wird ein Beispiel-Dialog gezeigt. Die vom Benutzer bzgl. der bereits getroffenen Entscheidung gestellte 'Warum'-Frage (etwa warum der Antrag nicht bewilligt wurde oder anders ausgedrückt was gegen eine Bewilligung spricht) führt zu einem 'Warum-Knoten', der die Ablehnung unterstützende Argumente enthält. Der Benutzer kann innerhalb dieses Kontext-Knotens navigieren, aber auch weitere Fragen bzgl. ausgewählter Konzepte stellen, wie z.B. daß er wissen möchte, wie der Umsatz (*turnover*) berechnet wurde. Eine solche Frage führt zu einem 'Wie-Knoten', der die Berechnungsformeln in Form eines Baums präsentiert. Zu den dargestellten Konzepten können Verständnisfragen gestellt werden, wie bezgl. des Umsatzkorrektors (*turnover corrector*). Die Erklärungen auf solche Fragen erfolgen in 'Was-Knoten', die z.B. die verbale Beschreibung eines Konzeptes oder auch statische, unbeeinflussbare Erklärungstexte enthalten. In diesen textuellen Beschreibungen können weitere Konzepte vorkommen (z.B. *turnover in goods*), die, wie in Hypertexten üblich, zu weiteren Knoten führen.

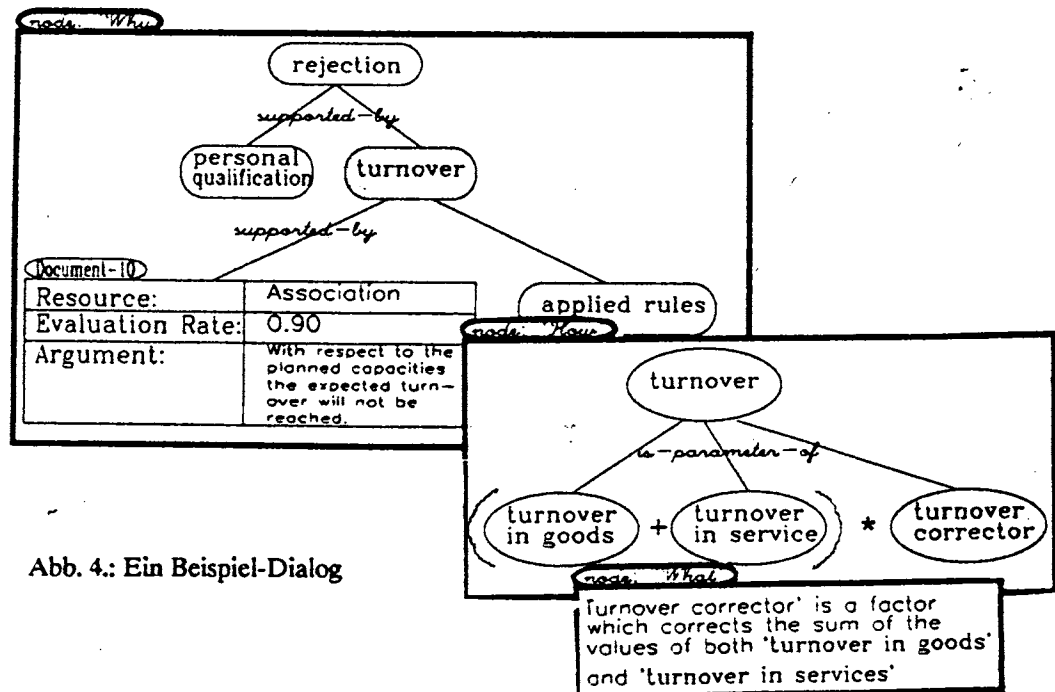


Abb. 4.: Ein Beispiel-Dialog

In den Kontext-Knoten können jeweils die Konzepte, auf die sich Fragen beziehen, als *Einstiegsknoten* in den entsprechenden Kontext-Knoten betrachtet (z.B. das Konzept 'turnover' in 'Wie-Knoten') und diese wiederum als Orientierungspunkte markiert werden, um den Benutzer wieder zurückzuführen (wie beim Backtracking). Darüberhinaus sollte die Möglichkeit gegeben sein, daß der Benutzer während des Navigierens für ihn relevante Gründe (in diesem Fall Knoten und Verknüpfungen) markiert, welche er am Ende des Dialogs zusammengestellt haben möchte⁶.

5. Zusammenfassung

Im Gegensatz zum nicht unterstützten Entscheidungsprozeß, bei dem der Sachbearbeiter unter Zeitdruck schon bei der Informationsaufbereitung auf seine Intuition und auf Heuristiken, die insbesondere bei mangelnder Erfahrung in der Auseinandersetzung mit einem Problem fehlerträchtig sind, angewiesen ist, werden hier seine Intuition und Erfahrung durch die hypertextuelle Organisation und die flexible Aufbereitung bzw. Präsentation der Informationsgrundlagen zuverlässig unterstützt.

Die Hypertext-Komponente vom WISKREDAS verwendet Hypertext-Techniken, ist jedoch keine echtes Hypertext-System. Eine Koppellung von WISKREDAS mit einem kommerziell verfügbaren Hypertext-System würde sicherlich einen Gewinn für WISKREDAS bedeuten, ist aber nur mit einem solchen Hypertext-System möglich, das wie WISKREDAS auf einem UNIX-Rechner ablauffähig ist, und für das eine entsprechende Schnittstelle zur Anbindung an WISKREDAS existiert. Die Systemleistung, eine schnelle und automatisierte Informationsbeschaffung, läßt sich durch die Hilfe des Hypertextes, die insbesondere bei der Präsentation zum Tragen kommt, ergänzen. Vor allem können beiseite gelegte Fälle jeweils über ein Bottom angesprochen und parallel (in verschiedenen Fenstern) bereitgehalten werden. Damit würde auch die Kosten für das zeitaufwendige Wiederfinden bzw. die Wiederaufnahme der Fälle reduziert werden.

Literatur

- Conklin, J. (1987): Hypertext - An introduction and a survey. IEEE Computer, Sept., 18-41.
- Dambon, P. (1988): Wissensbasierte Verwaltung eines administrativen Fallbearbeitungsvorgangs: Wissensrepräsentation in der "Fallbasis" von WISKREDAS. Universität Konstanz, Bericht SFB 221/B3-3/88.
- Dambon, P.; Glaser, F.; Kuhlen, R.; Thost, M. (1989): WISKREDAS: Ein Wissensbasiertes Kreditabsicherungssystem. Universität Konstanz, Bericht SFB 221/B3-3/89.
- Dambon, P. / Yetim, F. (1990): Integration of Hypertext into a Decision Support System. In: Herget, J.; Kuhlen, R. (eds.): Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen. Konstanz: Universitätsverlag, 64-77.
- Garg, P.K. / Scacchi, W. (1988): Composition of Hypertext Nodes. In: Proceedings Online Information'88, 63-73.

⁶ Dieser Aspekt ist insofern wichtig, als der Benutzer die Erklärung gegenüber anderen als Rechtfertigung verwenden will

Glaser, F. (1990): WIREMAN: Ein wissensbasiertes System zur Erarbeitung von Information aus Datenbanken für die Kreditwürdigkeitsprüfung von Unternehmensgründungen. In: Herget, J.; Kuhlen, R. (eds.): Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen. Konstanz: Universitätsverlag. 99-120.

→ Halasz, F. G. (1988): Reflections on Notecards: seven Issues for the next Generation of the Hypermedia Systems, in: Communications of the ACM, July 1988, Vol. 31, Nr. 7, 836-852.

Kuhlen, R. (1989): Rahmenbedingungen der Akzeptanz für den Einsatz wissensbasierter Verfahren. Workshop "Entscheidungsunterstützung in der öffentlichen Verwaltung - Entmythologisierung von Expertensystemen", Lüneburg, Juni 1989.

→ Kuhlen, R. (1991): Hypertext - ein nicht-lineares Medium zwischen Buch und Wissensbank. Berlin et al: Springer Verlag.

Thost, M. (1990): Informationsquellenmodelle. Ein Konzept zur wissensbasierten Bewertung der Glaubwürdigkeit von Meinungen. Konstanz: Martin Thost Verlag.

Wolf, M.F. (1988): Kooperative Entscheidungsunterstützung und Wissensverwaltung bei problematischen Finanzierungsentscheidungen: Die Entscheidungskomponente von WISKREDAS. Universität Konstanz, Bericht SFB 221/B3-2/88.

Yetim, F. (1989): Ein intertextuelles Hypertextmodell als Weiterentwicklung eines Volltextanalysesystems. In: Proc. Deutscher Dokumentartag 89; Informationsmethoden: Neue Ansätze und Techniken. 197-212.

→ Yetim, F. (1991): Eine Hypertext-Komponente zu einem Expertensystem: Benutzerfragen für Erklärungsdialoge. In: Maurer (ed.): Hypertext/Hypermedia '91, Berlin et al.: Springer, 27.-28.

Yetim, F.; Dambon, P. (1991): Can Hypermedia improve the Acceptance of Knowledge-Based Systems. In: Bullinger, H.-J. (ed.): Proceedings of the 4th International Conference on Human-Computer-Interaction, Stuttgart, Sept. 1991

HyperXPert

Hypertext und Expertensysteme

Annette Meinel
Universität Karlsruhe
Institut für Logik, Komplexität und Deduktionssysteme

Gliederung

- 1 Motivation
- 2 HyperXPert
 - 2.1 Statische Fenster
 - 2.2 Dynamische Fenster
 - 2.3 Navigationshilfsmittel
 - 2.4 Ausgabefunktion
- 3 Zusammenfassung

1 *Motivation*

Ausgangspunkt der Entwicklung von *HyperXPert* war die Expertensystem-Shell *D3*. Mit dieser Shell können Expertensysteme für diagnostische Problembereiche erstellt werden.

Zur Problemklasse „Diagnostik“ gehören alle Aufgabenbereiche, bei denen die Lösungen aufgrund von Problemmerkmalen ausgewählt werden können, wie z.B. Krankheiten bei der medizinischen Diagnostik, defekte Bauteile bei technischen Geräten oder Empfehlungen bei der Produktauswahl aus einem Katalog.

Das in einem Expertensystem gespeicherte Wissen wird zur Problemlösung von Fällen verwendet. Dabei wird nur ein Teil des Wissenspotentials genutzt, nämlich der Teil, der für die Problemlösung relevant ist. Die ebenfalls vorhandene Erklärungsfähigkeit des Wissens wird nur minimal berücksichtigt und ausgewertet. Um dieses Potential, z.B. zur Schulung, nutzen zu können wurde *D3* um das Hypertext-System *HyperXPert* und eine tutorielle Komponente erweitert.

Angenommen ein Student möchte sich umfassend in die Reparaturdiagnostik des Automotors einarbeiten. Dazu würde man ihm nicht nur heuristische Regeln mitteilen, sondern auch Hintergrundwissen über die Funktionsweise der Bauteile und über Auswirkungen typischer Defekte vermitteln, Fallbeispiele erläutern, teilweise auch Apriori-Häufigkeiten und statistische Korrelationen angeben, sowie durch allgemeines, informelles Wissen über den Automotor ergänzen.

Diese Vielfalt an Wissensformen, die sich teilweise überschneiden und teilweise ergänzen, sind aus zwei Gründen auch für Expertensysteme wichtig. Zum einen wird damit die Problemlösungsfähigkeit erhöht, da die verschiedenen Ausprägungen des Wissens zur gegenseitigen Überprüfung der Lösungen herangezogen werden können. Und zum anderen können sie beim tutoriellen Einsatz verschiedene Sichten des Anwendungsgebiets vermitteln und zur Vertiefung des Lernstoffes dienen. Wissen läßt sich in folgenden Ausprägungen darstellen:

- *Erfahrungswissen*

Meist heuristisches Wissen, das die Beziehung von Symptomen zu Diagnosen in Form von heuristischen Regeln beschreibt.

- *Modellbasiertes (kausales) Wissen*
Wissen über allgemeine Zusammenhänge zwischen Diagnosen und Symptomen. Es kann dabei sowohl das Verhalten der einzelnen Bauteile im Normal- und Fehlerzustand, als auch die Fehlerfortpflanzung im Modell beschrieben werden.
- *Statistisches Wissen*
Unsicheres Wissen, das durch statistische Aufbereitung einer Menge gelöster Fälle gewonnen wird.
- *Fallvergleichendes Wissen*
Wissen über erfolgreich gelöste Fälle, das zu Analogieschlüssen herangezogen wird.
- *Allgemeines Wissen*
Wissen das sich nur schlecht formalisieren läßt, z.B. Bilder oder Übersichten.

Abb. 1 zeigt die verschiedene Wissensarten, die in D3 und HyperXPert repräsentiert werden können. Dadurch ist die Funktionalität und Aussagekraft eines Expertensystems um Hypertext-Aspekte erweitert worden, so daß es jetzt nicht nur zur Problemlösung, sondern auch zur Schulung eingesetzt werden kann.

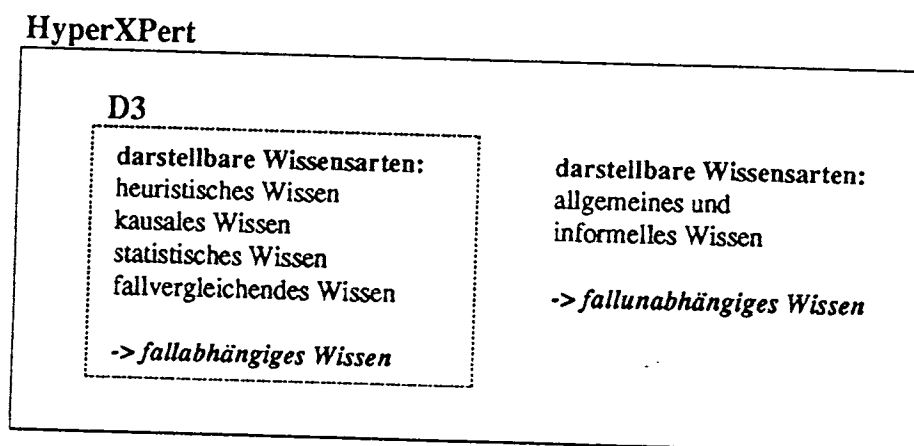


Abb. 1: Darstellbare Wissensarten in D3 und HyperXPert

Werden Expertensysteme mit Hypertext-Dokumentationen zu Lernzwecken eingesetzt, so haben sie gegenüber anderen Wissensmedien, wie z.B. Fachbüchern, zwei Vorteile:

- *Größere Anpassungsfähigkeit an Benutzerwünsche*

Bücher sind sequentielle Wissensmedien, die man am bequemsten in der vorgegebenen Reihenfolge liest. Auch ein Eingehen auf das unterschiedliche Vorwissen oder die unterschiedlichen Interessen verschiedener Lesergruppen ist kaum möglich. Dagegen kann das Wissen in hypertext-orientierten Expertensystemen nach verschiedenen Kriterien aufbereitet, assoziativ gelesen und vor allem auf konkrete Problemfälle angewendet werden.

- *Überprüfbarkeit des Wissens*

Der entscheidende Vorteil aber ist, daß das Wissen nicht nur von anderen Menschen, sondern auch vom Wissensmedium selbst interpretiert und durch Anwendung auf bekannte Fälle überprüft werden kann.

2 *HyperXPert*

HyperXPert ist ein Hypertext-System mit dem Hypertext-Dokumentationen erstellt werden können die mit Expertensystemen, die mit der Expertensystem-Shell D3 erstellt worden sind, zusammenarbeiten können. Abb. 2 zeigt die allgemeine Struktur einer Hypertext-Dokumentation von HyperXPert mit verschiedenen Fenstern, assoziativen Verweisen, Fußnoten und Menüpunkten.

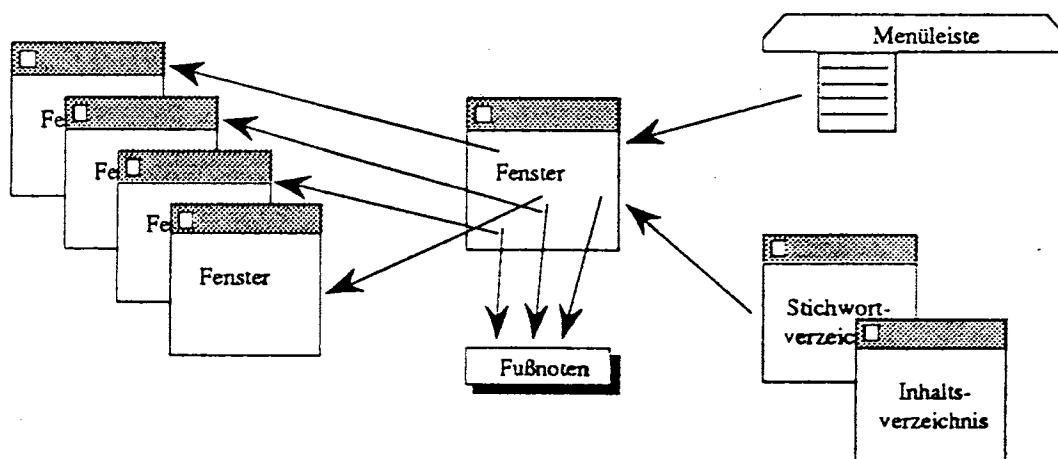


Abb. 2: Allgemeine Struktur einer Hypertext-Dokumentation von HyperXPert

Um die für Lern- und Schulungszwecke benötigten verschiedenen Wissensarten repräsentieren zu können, stellt HyperXPert dem Autor einer Dokumentation drei verschiedene Fenstertypen zur Verfügung:

- *Statische Fenster* zur Darstellung von fallunabhängigem Wissen, d.h. allgemeinem oder informellem Wissen das nicht von einem bestimmten Fall und den dazugehörigen Symptomen und Diagnosen abhängt.
- *Dynamische Fenster* zur Darstellung von fallabhängigem Wissen, d.h. formalisiertem Wissen des Expertensystems, das erst durch die Symptome und Diagnosen eines Falles aussagekräftig wird.
- *Fußnoten* zur Darstellung von Zusatzwissen.

Das Stichwort-, das Inhaltsverzeichnis und die Menüleiste sind Navigationshilfsmittel, die dem Leser helfen sollen, sich in einer großen Hypertext-Dokumentation zurechtzufinden und die gezielte Suche nach Informationen erleichtern. Die Menüpunkte der Menüleiste dienen außerdem als Einsprungpunkte in die Hypertext-Dokumentation.

2.1 *Statische Fenster*

In einem statischen Fenster kann fallunabhängiges Wissen, d.h. allgemeines oder informelles Wissen; dargestellt werden.

Jedes statische Fenster besteht aus zwei Teilen, dem Hintergrund und dem Vordergrund (s. Abb. 3). Der Hintergrund kann leer sein, ein Bild, Text oder Bild und Text kombiniert enthalten. Der Vordergrund setzt sich aus maussensitiven Bereichen (Tasten, Textfelder, Symbole oder Pop-Up-Menüs) zusammen, durch die einzelne Textabschnitte oder Bildteile des Hintergrundes mit anderen Fenstern der Hypertext-Dokumentation verbunden werden können. Dadurch wird eine assoziative Verknüpfung mit anderen Fenstern der Dokumentation aufgebaut.

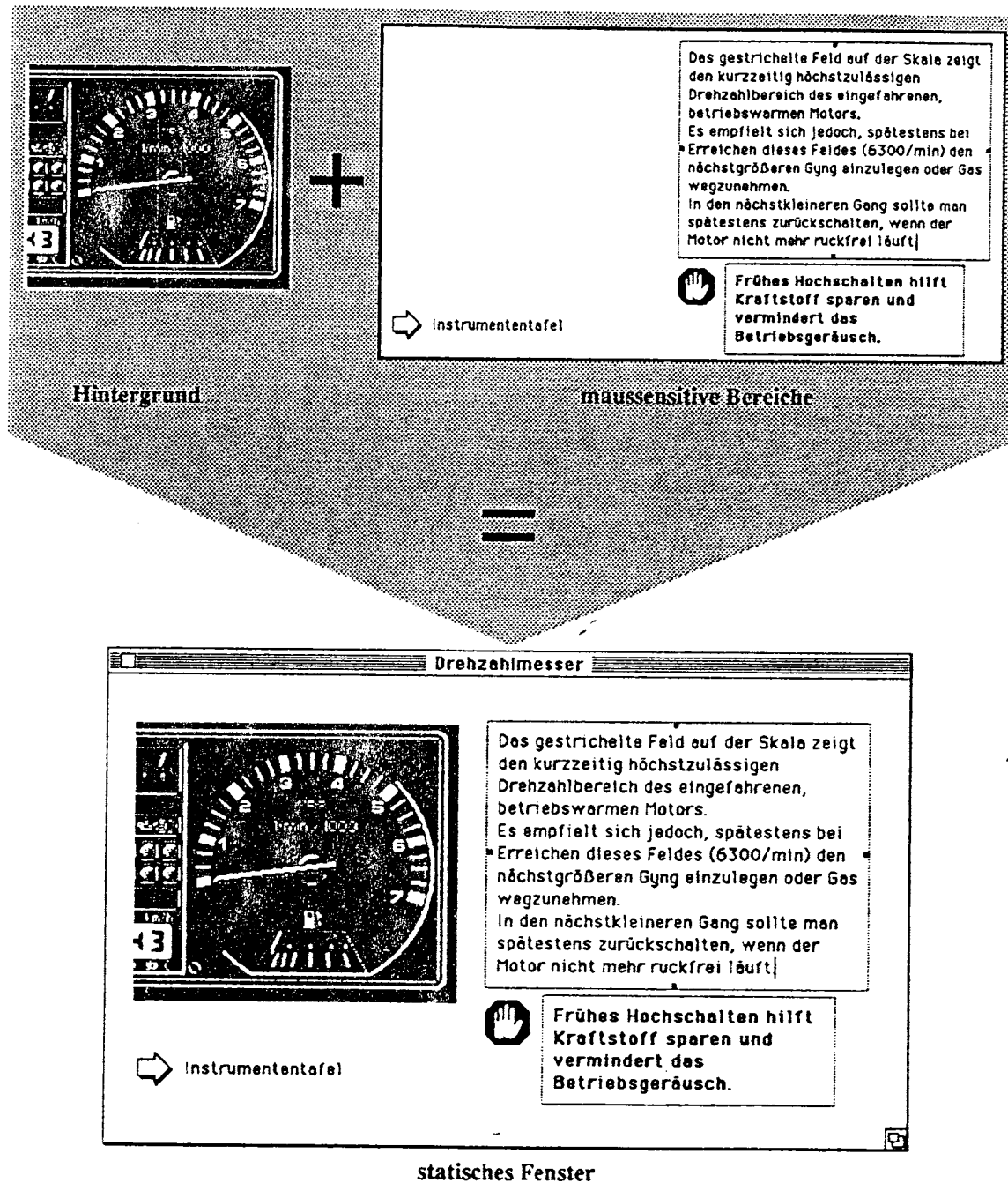


Abb. 3: Prinzipieller Aufbau eines statischen Fensters

2.2 Dynamische Fenster

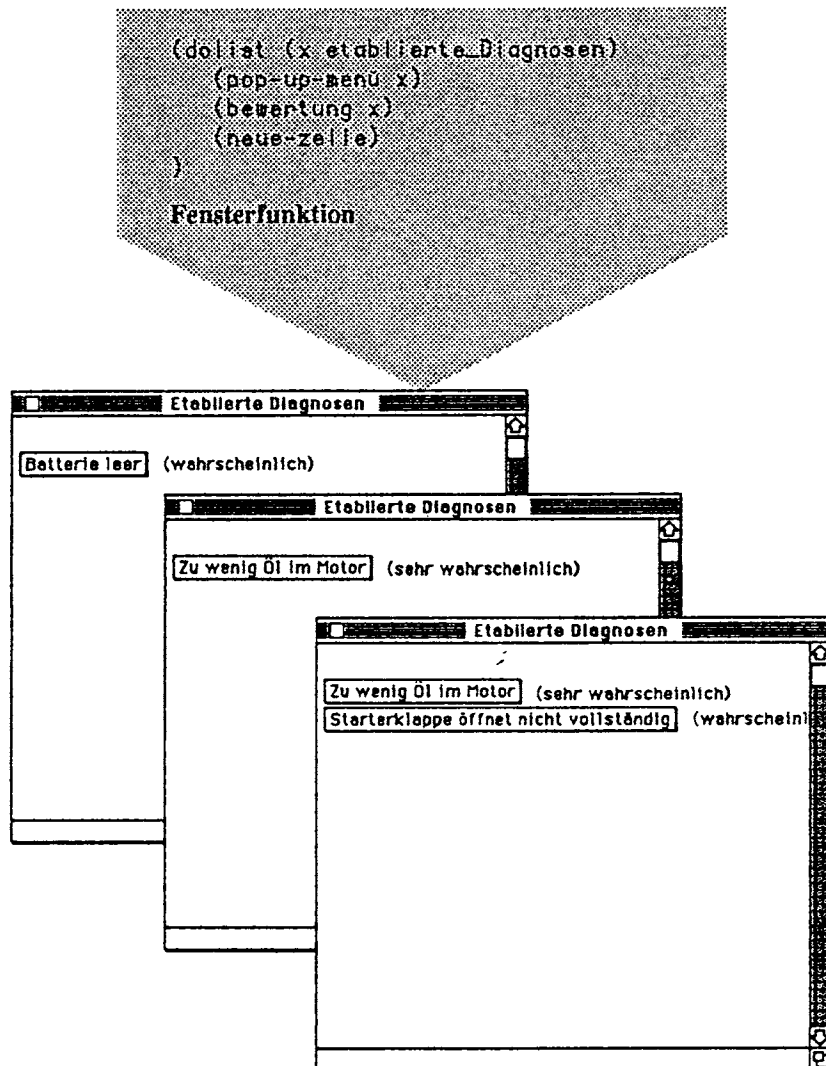


Abb. 4: Prinzipieller Aufbau eines dynamischen Fensters

Der Fensterinhalt und das Aussehen eines dynamischen Fensters kann nicht wie bei den statischen Fenster explizit angegeben werden, da in ihm fallabhängiges Wissen dargestellt wird. Die Wissensdarstellung ist abhängig von dem gerade bearbeiteten Fall und den dazugehörigen Symptomen und Diagnosen. Darum wird der Fensterinhalt durch LISP-Funktionen und spezielle Schnittstellenfunktionen spezifiziert (s. Abb. 4). Genau wie bei den statischen Fenstern können auch dynamische Fenster assoziativ, durch maussensitive Bereiche, mit anderen Fenster der Hypertext-Dokumentation verbunden werden.

2.3 Navigationshilfsmittel

Die von HyperXPert bereitgestellten Navigationshilfsmittel sollen zum einen verhindern, daß sich ein Leser in einer großen Hypertext-Dokumentation verirrt und zum anderen die gezielte Suche von Informationen erleichtern. HyperXPert stellt vier verschiedene Mechanismen zur Verfügung:

- *Menüs und Menüpunkte*

Zu jeder Hypertext-Dokumentation können Menüs und Menüpunkte generiert werden. Mit diesem Mechanismus werden Einsprungmöglichkeiten in die Dokumentation definiert.

- *Stichwortverzeichnis*

In diesem Mechanismus können frei wählbare Stichworte mit einzelnen Fenstern der Hypertext-Dokumentation verbunden werden. Dadurch wird die gezielte Informationssuche erheblich erleichtert (s. Abb. 5).

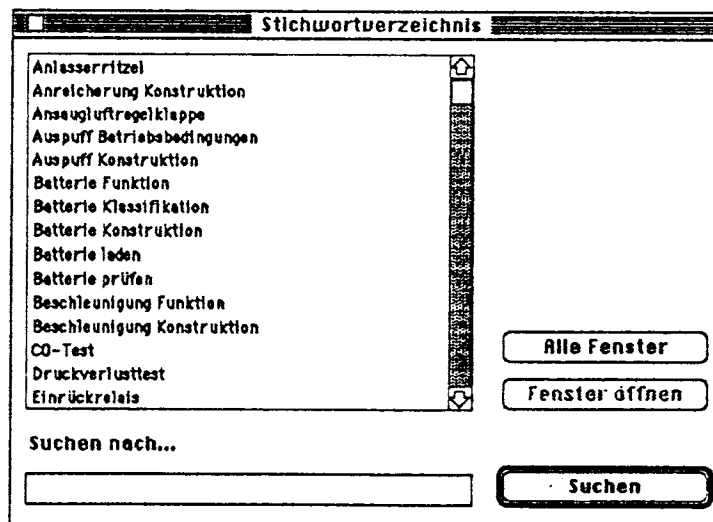


Abb. 5: Stichwortverzeichnis einer Hypertext-Dokumentation

- *Lineare Strukturen anlegen*

Die einzelnen Fenster einer Hypertext-Dokumentation sind assoziativ, d.h. nicht linear, untereinander verknüpft. Es kann aber in manchen Fällen vorteilhaft sein, zusätzlich zu dieser Struktur auch noch eine lineare Struktur unter die Dokumentation zu legen, so daß man vorwärts und rückwärts blättern kann. Dies

ist sowohl für den Leser als auch für den Autor einer Hypertext-Dokumentation sinnvoll. Ein Leser kann sich so sehr leicht einen Überblick über das in der Dokumentation gespeicherte Wissen verschaffen und der Autor kann seine Dokumentation bei der Erstellung auf Vollständigkeit prüfen.

HyperXpert bietet zusätzlich noch die Möglichkeit mehrere lineare Strukturen für eine Hypertext-Dokumentation zu definieren. So können z.B. die unterschiedlichen Vorkenntnisse von Lesern (Anfänger, Fortgeschrittene, Experten) berücksichtigt werden.

- *Inhaltsverzeichnis*

Im Inhaltsverzeichnis kann eine Hypertext-Dokumentation und ihre zugrundeliegenden linearen Strukturen zu Oberbegriffen zusammengefaßt werden. Mit diesem Mechanismus wird es dem Leser erleichtert, sich einen Überblick über die Dokumentation zu verschaffen.

2.4 *Ausgabefunktion*

Um mit einer Hypertext-Dokumentation wirklich sinnvoll arbeiten zu können ist auch die Erstellung eines Ausdrucks wichtig. Dazu muß die Dokumentation vorher geeignet aufbereitet werden.

In HyperXPert wird mit dem Inhaltsverzeichnis begonnen, dann werden die einzelnen Fenster mit Hilfe der linearen Struktur aneinander gehängt und mit Überschriften und Seitenzahlen versehen. Die Ausgabe erfolgt nicht direkt auf einen Drucker sondern wird im Format eines beliebigen Textverarbeitungsprogrammes abgespeichert, so daß die Ausgabe sehr leicht geändert und weiterverwendet werden kann.

3 *Zusammenfassung*

Natürlich kann ein Expertensystem und eine Hypertext-Dokumentation ein Lehrbuch nur ergänzen und nicht ersetzen. Die Vorzüge des Lehrbuchs liegen in dem Unterschied der natürlichen Sprache gegenüber einem Formalismus. Die natürliche Sprache ist angenehmer zu lesen und vor allem ausdrucksstärker, d.h. sie umfaßt wesentlich mehr sprachliche Nuancen als z.B. in einem Expertensystem dargestellt werden können. Für einen Schüler kann jedoch ein strenger Formalismus zu Anfang sehr hilfreich bei der Konzentration auf das Wesentliche sein, obwohl er ihn irgendwann transzendieren muß.

Durch die Möglichkeit in der Hypertext-Dokumentation einfach zwischen dem fallabhängigen, formalisierten Wissen des Expertensystems und dem fallunabhängigen, informellen Wissen einschließlich Bildern aus Lehrbüchern direkt hin und herspringen zu können, kann der Schüler die Vorzüge beider Mechanismen kombinieren.

Besonders attraktiv ist dies, wenn die Schüler ihr erlerntes Wissen selbständig in einer simulierten Umgebung erproben können. Dafür wurde der Kritikmodus entwickelt. Im Kritikmodus löst der Schüler reale oder von einer Falldatenbank vorgegebene Fälle selbst und wird vom Expertensystem kritisiert. Dabei kann nicht nur das Endergebnis kommentiert werden, sondern auch der Diagnostikprozeß, d.h. die zwischenzeitlichen Verdachtsdiagnosen und die zusätzlichen Tests, die der Schüler zur Abklärung der Enddiagnosen verwendet hat.

Das Projekt EbsKart - Medienmix in einem interaktiven Auskunftssystem

Paul Ferdinand Siegert

Martin Warnke

Walter Uka

Universität Lüneburg, Juli 1991

Im folgenden berichten wir von einem Museums-Auskunfts-System zu einem komplexen Bild-Text-Dokument des hohen Mittelalters.

Im ersten Teil (M.W.) werden der abgebildete Gegenstand und die technischen Methoden der Abbildung thematisiert, der zweite Teil (P.f.S.) handelt von wünschbaren und teilweise vorhandenen Hypersystem-Eigenschaften, der dritte und letzte (W.U.) von den Besonderheiten, die im Zusammenhang mit eingebundenen Video-Bildern entstehen.

Teil 1: Eine interaktive Monographie: das Projekt EBSKART

Seit Ende 1989 verfolgen wir an der Universität Lüneburg das Arbeitsvorhaben, zu einem komplexen mittelalterlichen Dokument ein Erklärungswerk anzufertigen [5][6]. Bei diesem Unterfangen kommt uns die eigentümliche, schon an Hypersysteme erinnernde Struktur des Gegenstandes entgegen:

Das älteste Hyperdokument der Welt

Zwischen 1230 und 1300 [3] ist im norddeutschen Raum ein Kunstwerk geschaffen worden, das mit seiner engen Verknüpfung zwischen Text und Bild ein Kompendium mittelalterlichen Weltwissens darstellt. Gemeint ist die Ebstorfer Weltkarte [2], eine Darstellung der dem hohen Mittelalter bekannten Welt, die zum Genre der *map-pae mundi* gehört.

Auf einer *mappa mundi* hängt auf engstem Raum alles mit allem zusammen: Orte, Begebenheiten und Mythen, lexikalisches Wissen und die Lehren der Religion formen ein hermetisches Ganzes, ein Welt-Bild und eine Welt-

Anschauung.

Die Informationspartikel sind nicht, wie nach dem Durchbruch des Buchdrucks üblich, linear strukturiert, sondern liegen alle miteinander auf einer Fläche beachtlichen Ausmaßes: auf dreieinhalb mal dreieinhalb Metern im Quadrat sind die Welt und die sie kommentierenden Texte und Bilder gemeinsam und in großer Detailfülle versammelt. Damit ist die Ebstorfer Weltkarte die größte uns im Detail bekannte mittelalterliche Karte überhaupt [3] (siehe Abb 1: Ebstorfer Weltkarte gesamt).

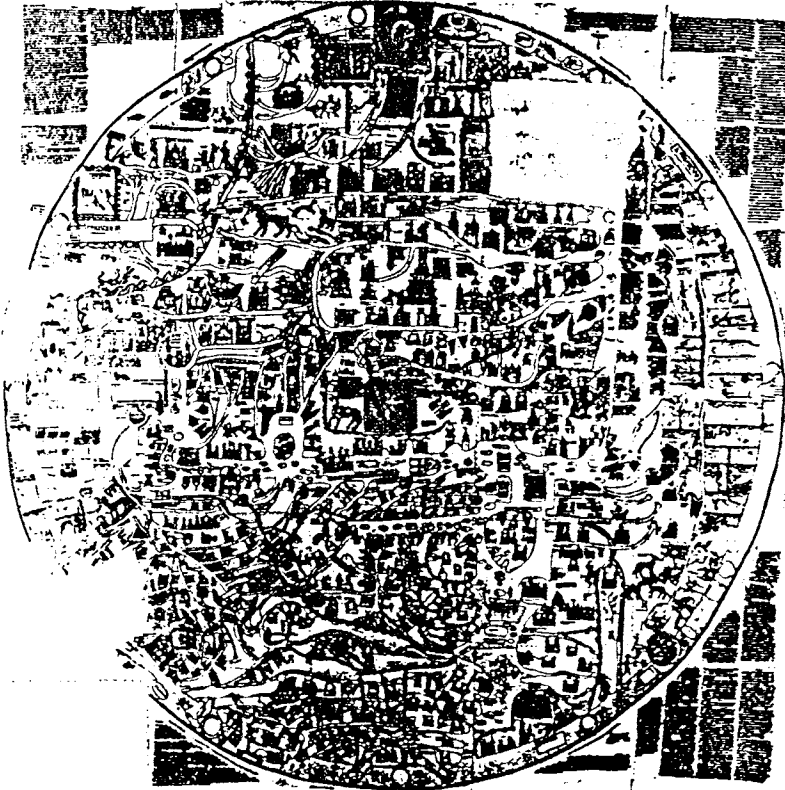


Abbildung 1: Ebstorfer Weltkarte gesamt

Die Ebstorfer Weltkarte nutzt die synergetischen Wirkungen einer rhetorisch geschickten Kombination von Bild und Text absichtsvoll aus. In diesem Sinne kann sie durchaus als ein frühes Beispiel für eine Hyperdokument-Struktur aufgefaßt werden, allerdings auf konventionellem Träger (Schafspergament [2]) und nicht in einzelne bildschirmgroße Ausschnitte zerteilt.

Eher noch als manche Produkte multimedialer Artistik neueren Datums bringt sie zur Geltung, was mit Hypersystemen zu erreichen ist: eine der rhetorischen Bauart (siehe dazu [1]) des Materials angemessene Darstellung durch flexiblen Einsatz mehrerer Medien.

Der Gehalt der Ebstorfer Weltkarte

Das auf der Ebstorferkarte dargestellte Erdenrund teilt sich in die Kontinente Asien, das die obere Hälfte bedeckt, Europa in dem unteren linken Viertel und Afrika unten rechts.

Rund 500 Gebäudedarstellungen stellen Ortschaften dar, etwa 160 Gewässerdarstellungen stehen für Meere, Flüsse und Seen, 60 Inseln und Gebirgszüge vervollständigen den Kanon geographischer Objekte. 45 Menschen oder menschenähnliche Wesen stehen für Völker, biblische oder mythologische Begebenheiten, etwa 60 Tiere geben einen Überblick zum damaligen zoologischen Wissen, das auf der Seite der Fauna durch etliche Pflanzen ergänzt wird.

Diese Bildeintragungen werden meist von Textpassagen begleitet oder auf den Kartenrändern, die von dem

Erdrund nicht bedeckt werden, ausführlich kommentiert.

Insgesamt gibt es etwa 1500 einzelne Textpassagen, oft lediglich kurze Ortsbezeichnungen, aber auch längere Ausführungen zu biblischen, mythologischen, zoologischen oder geographischen Themen.

Als Beispiel für den Bereich Zoologie möchte ich einen kurzen Text anführen, der direkt neben einer bildlichen Darstellung zu finden ist (Abb. 2):

»Die Antilope ist ein äußerst flinkes Tier, so flink, daß kein Jäger sich ihm nähern kann. Es hat lange, sägeförmige Hörner und schneidet damit hohe Bäume ab. Doch beim Hinaufsteigen bleibt es oft mit den Hörnern im Gezweig hängen, und wenn es dann nicht mehr freikommt, so heult es laut auf. Dies hört der Jäger und fängt das Tier« [4].



Abb. 2: Die Antilope, der zugehörige Text und der Turm zu Babel

Geographisch am detailliertesten ist der norddeutsche Raum dargestellt. So ist der ansonsten völlig unbedeutende Ort Ebstorf, in dessen Kloster die Karte aufgefunden wurde, mit drei Märtyrergräbern verzeichnet; Lüneburg und Braunschweig sind vertreten, wie auch die kleiner dargestellten Städte Hannover und Bremen. Insgesamt ist die Überbetonung Norddeutschlands, speziell Lüneburgs, nicht zu übersehen, ein Argument für eine Urheberschaft der Karte im Lüneburger Raum.

Die geographische Sicht auf die Welt ist eng mit der christlichen Heilslehre verknüpft. Die Welt wird von Christus gehalten (siehe Gesamtansicht): das Haupt Christi befindet sich im Osten, woher das Heil kommt, die linke und rechte Hand Christi liegen im Norden und Süden, die Füße im Westen. Jerusalem ist wie bei allen mittelalterlichen mappae mundi exakt in der Mitte zu finden: als Nabel der Welt.

Das Paradies, direkt neben dem Haupte Christi (siehe Abbildung 3: das Paradies und das Haupt Christi) und von Hindernissen umgeben, wurde als wirklicher Ort auf der Erde gedacht. So heißt es in der Textpassage, die zum Paradies gehört:

»Asien ist nach einer Königin desselben Namens benannt. Seine erste Region von Osten her ist das Paradies, wahrhaftig ein angenehmer und wonnevoller Ort, den Menschen unzugänglich und mit einer himmelhohen Feuerwand umgeben. Darin befindet sich das Holz des Lebens, ein Baum, und wer von dessen Frucht isst, altert nicht und stirbt nie. Im Paradies entspringt eine Quelle, die sich in vier Ströme aufteilt; sie verschwinden innerhalb des Paradieses im Erdreich, doch treten sie weit weg in anderen Erdgegenden wieder an die Oberfläche. Und zwar entspringt der Physon, der auch Ganges heißt, in Indien vom Berg Orcobares und mündet, nachdem er ostwärts geflossen ist, in den östlichen Ozean. Der Geon sodann, der auch Nil heißt, wird bald, nachdem er beim Atlas-Gebirge entsprungen ist, wieder von der Erde aufgenommen und bricht

dann nach verborgenem Lauf in der Nähe des Roten Meeres nochmals hervor, strömt gemächlich durch Äthiopien und Agypten und wird, in sieben Mündungsarme unterteilt, bei Alexandria vom Mittelmeer aufgenommen* [4].



Abbildung 3: das Paradies und das Haupt Christi

Viele weitere biblische Begebenheiten, vor allem aus dem Alten Testament, werden gezeigt, so z. B. die Arche Noah nach ihrer Landung auf dem Berge Ararat, zusammen mit der Taube, die den Ölzweig im Schnabel hält und der Turm zu Babel.

Ein weiterer wichtiger dargestellter Gegenstandsbereich besteht aus den Legenden um Alexander den Großen. Begebenheiten aus seinem Leben sind u.a. in der Nähe des Hauptes Christi, also an exponierter Stelle, wiedergegeben.

Die technische Basis von EBSKART

Der Schwerpunkt der dokumentarischen Erschließung der Ebstorkarte liegt auf ihrer bildlichen Darstellung in Verbindung mit einer gut lesbaren Fassung der Texte. Für eine solche Aufgabenstellung sind HyperCard-ähnliche Programme am angemessensten, weil sie geeignete graphische Darstellungstechniken bieten, hinzu kommt die Programmierbarkeit, die bei EBSKART wegen der speziell entworfenen Bedienoberfläche eine große Rolle spielt. Die Entscheidung fiel dann auch auf HyperCard von Apple und den Apple Macintosh.

Ein topographisch orientiertes Zugriffsschema auf Daten

Die Ebstorkarte ist ein spezielles Beispiel für flächig angeordnete Datensammlungen, bei denen die gegenseitige Lage der Elemente eine wesentliche Rolle spielt. Bei solcherart strukturierten Daten bietet sich ein topographisches Ordnungs- und Zugriffsschema an, wie es bei üblichen Kartenwerken vorzufinden ist.

So haben wir uns dazu entschlossen, dem Paradigma eines Autoatlas mit Übersichts- und Detailkartendarstellungen verschiedenen Maßstabs zu folgen.

Auf oberster Ebene befindet sich die Gesamtübersicht der Ebstorkarte, eingeteilt in neun Felder (Abbildung 4). Statt den Betrachter den Maßstab der Darstellung explizit wählen zu lassen, haben wir uns für eine *point and click*-Technik entschieden: das Feld, das man vergrößert betrachten möchte, wird mit der Maus angeklickt.

Wichtig ist hierbei, daß der Benutzer des Systems in jedem Moment weiß, welche Operationen möglich sind.

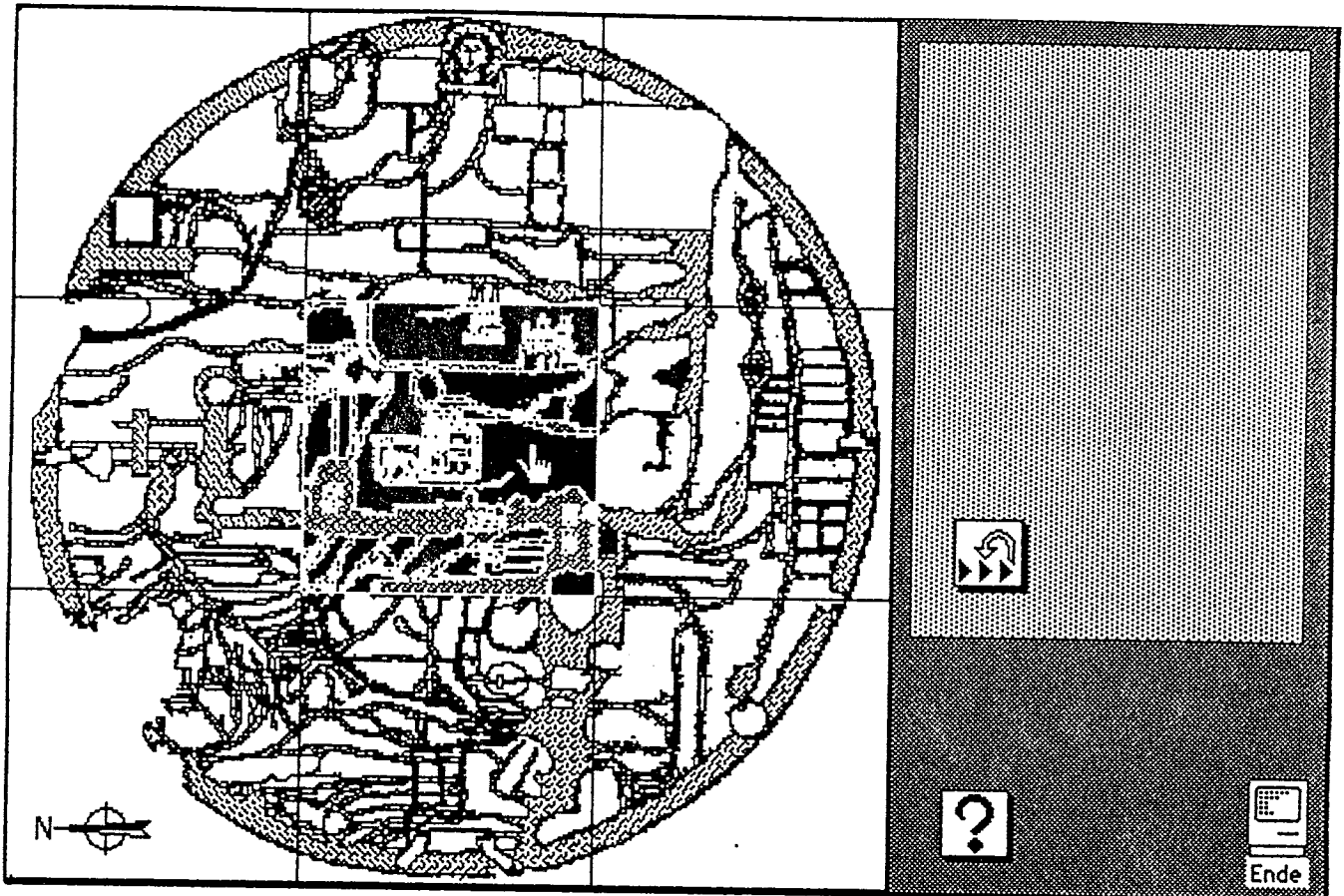


Abbildung 4: Die Übersichtskarte von EbsKart

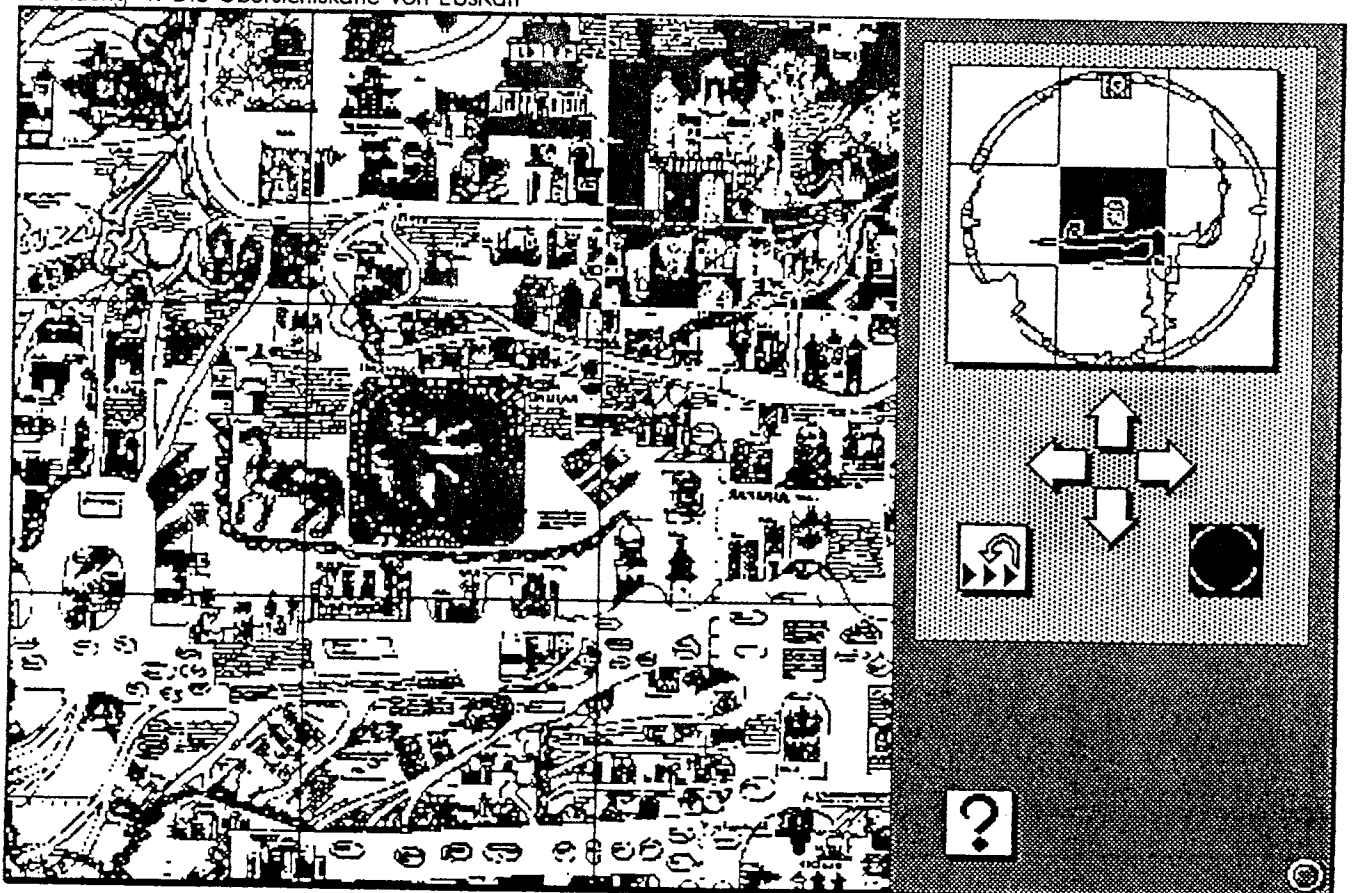


Abbildung 5: das mittlere Neunteil

Deshalb haben wir es durchgängig so realisiert, daß sich aktivierbare Elemente bei Berührung durch die Maus graphisch hervorheben und so signalisieren, daß eine Benutzeroperation möglich ist.

In Abb. 4 liegt der Mauszeiger in Form einer Hand über dem mittleren Neuntel der Gesamtdarstellung. Ein Mausklick bringt dieses Neuntel auf den Bildschirm (Abbildung 5).

Auf dieser Maßstabsebene befinden sich nun wieder je neun anwählbare Segmente, von denen dasjenige oben rechts hervorgehoben dargestellt wird, weil der Mauszeiger darauf weist. Ein Mausklick verzweigt erwartungsgemäß auf dieses Segment, doch bevor das geschieht, sollen die anderen Bedienelemente besprochen werden.

Die Funktion der windrosenförmig angeordneten vier Pfeile auf der rechten Seite des Bildschirms ist es, in die entsprechende Richtung zu blättern, und zwar auf derselben Maßstabsebene. Ein direkter Sprung zu jedem der Segmente der gleichen Maßstabsebene ist mit dem Block von neun Feldern möglich, der oben auf der rechten Seite angebracht ist und ein stark verkleinertes Abbild der Ebstorkarte trägt.

Dieser Block verzweigt per Mausklick auf den entsprechenden Abschnitt und zeigt gleichzeitig den Standort des Betrachters an. Diese Zusammenlegung von Aktion und Rückmeldung ist nach unseren Erfahrungen wesentlich dafür, daß Benutzer schnell mit der Bedienschnittstelle des Systems zurecht kommen.

Links unter der Windrose liegt ein für alle Hypersysteme übliches Bedienelement, der Ariadnefaden, der den Benutzer um je einen Bildschirm zurück führt. Rechts daneben liegt ein Symbol, das für den Wechsel auf die nächstgrößere Maßstabsebene zuständig ist; man beachte auf den folgenden Abbildungen der feineren Maßstäbe, daß das Symbol sein Aussehen ändert und jeweils denjenigen Bruchteil der Gesamtkarte hervorhebt, auf den es verzweigt, hier also das ganze Rund, auf der folgenden Maßstabsebene nur noch das mittlere Neuntel.

Verzweigt man nun per Mausklick auf das obere rechte Neuntel der Karte von Abbildung 5, so gelangt man zum Ausschnitt, der auf Abbildung 6 gezeigt wird:

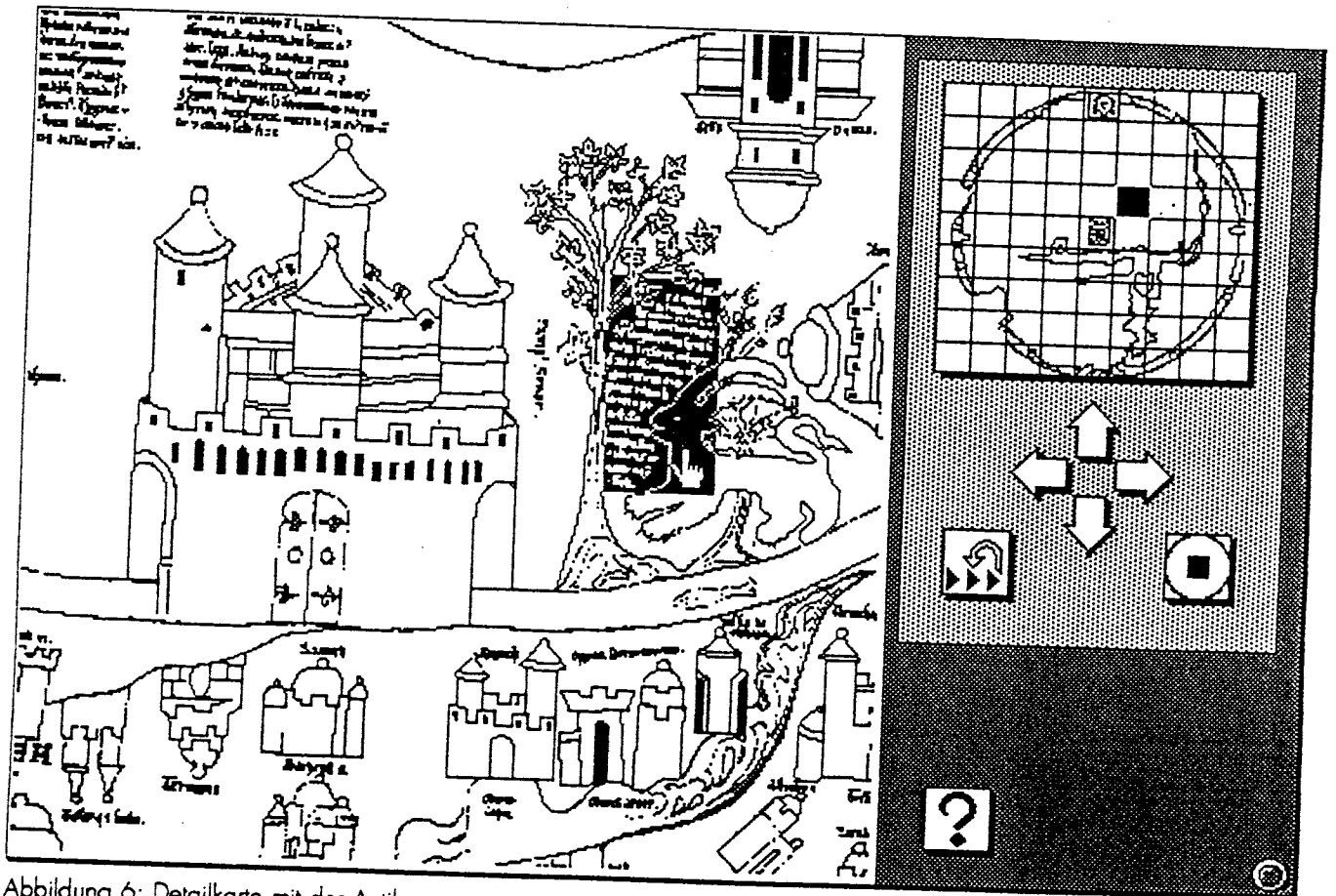


Abbildung 6: Detailkarte mit der Antilope

Dieses ist die detaillierteste Maßstabsebene und umfaßt insgesamt 81 Bildschirme. Sie sind wieder im Navigations- und Standortanzeigeblock oben rechts verzeichnet; wir befinden uns offensichtlich in der sechsten Spalte von links, vierte Zeile von oben. Ein Maßstabswechsel auf die höhere Ebene bringt uns in das mittlere Neuntel; das Symbol rechts unterhalb der Windrose zeigt dies an.

Der Kartenausschnitt bildet unter anderem die Antilope ab, von der schon weiter oben die Rede war. Neben ihr steht der Text im Faksimile, der hervorgehoben dargestellt ist, weil der Mauszeiger über ihm steht. Auf dieser Maßstabsebene sind sämtliche Texte der Ebstorkarte anwählbar. Ein Mausklick bringt uns zu der zugehörigen Textkarte (Abb.: 7):

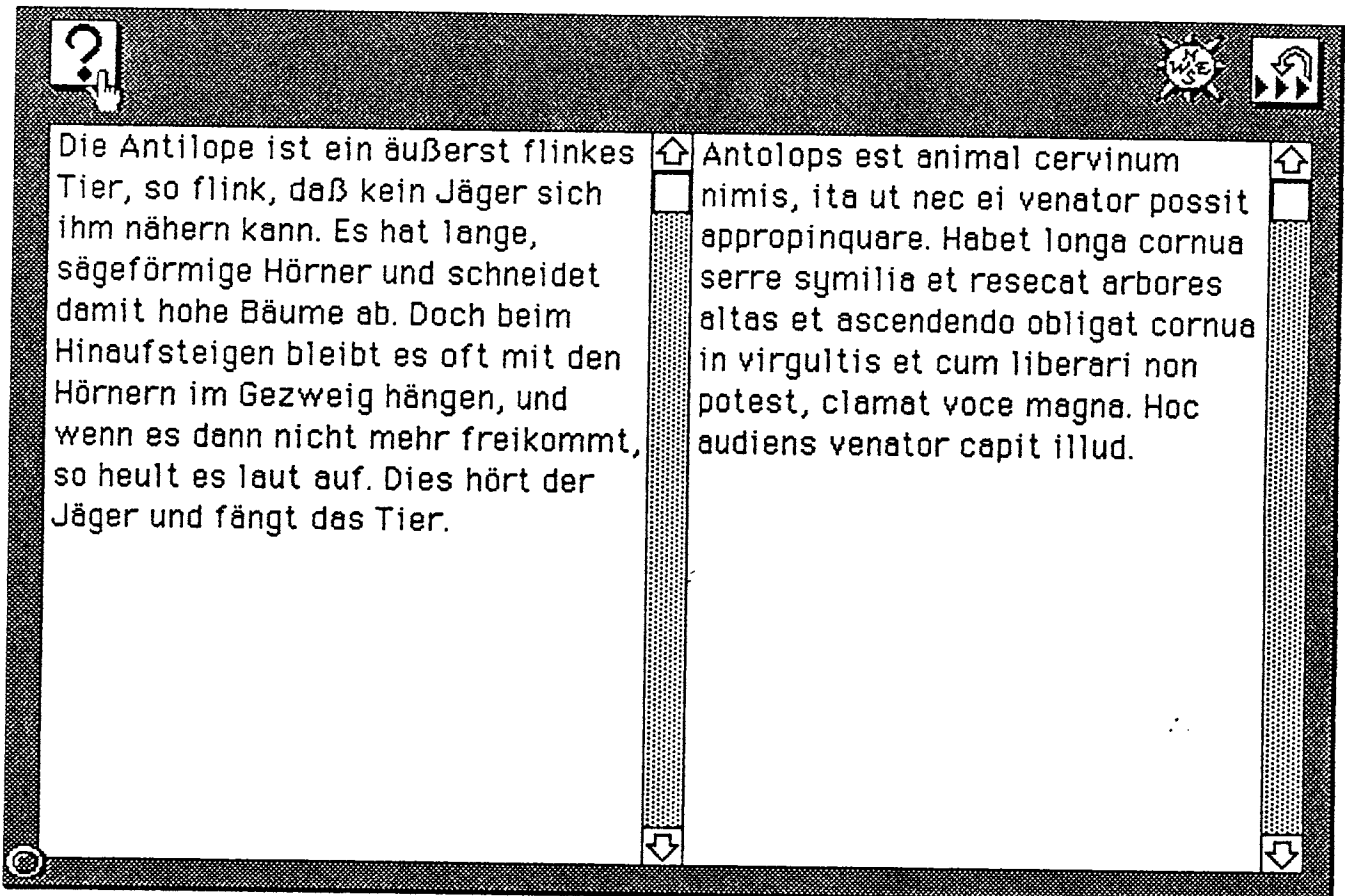


Abbildung 7: die Textkarte

Die 1677 Textkarten zeigen den lateinischen Text in lesbarer Form und eine deutsche Übersetzung [4], die uns freundlicherweise von Prof. Dr. Kugler (Kassel), einem Spezialisten für die Ebstorkarte, zur Verfügung gestellt worden sind.

Neben dem Ariadnefaden oben rechts gibt es ein Bedienelement, durch das man von der Textkarte auf das Atlasblatt kommt, auf der der Text verzeichnet ist. Die Stelle wird, wenn das Atlasblatt aufgeschlagen wird, durch Blinken hervorgehoben.

Oben links befindet sich, wie auch auf den Atlaskarten, das Bedienelement für die Volltextsuche. Klickt man es an, so kann man in einem Dialogfenster eine Zeichenkette eingeben, nach der in der Textbasis gesucht wird. So kann man leicht einzelne Stellen auf der Ebstorkarte auffinden: die Textsuche ermittelt die Textkarte, die Verzweigung auf das Atlasblatt springt auf die gesuchte Stelle.

In der nächsten Ausbaustufe von EBSKART wird ein Index die Suchtechniken abrunden. Das Werkzeug für die Indexierung ist gerade fertiggestellt worden.

Ausblick

Wir beabsichtigen, das System auf die wesentlichsten Bild- und Textdokumente des Mittelalters auszudehnen, die in unmittelbarem Zusammenhang mit der Ebstorkarte stehen. Eine solche wachsende Basis von Texten und Bildern erfordert eine leistungsfähigere Datenhaltung und -Archivierung, als sie mit HyperCard möglich wäre. Aus diesem Grunde experimentieren wir mit Archivierungssystemen, die Texte und Farbgraphiken effizient verwalten. Als Ziel solcher Entwicklungen sehen wir den Arbeitsplatz für Historiker und Kunstwissenschaftler, die über das Mittelalter arbeiten.

Teil 2: EBSKART - Probleme einer Hypersystemoberfläche im Kulturbetrieb

Während der Arbeit am Projekt EbsKart haben sich zahlreiche Probleme ergeben, von denen hier zwei angesprochen werden sollen. Der erste Punkt, „Personal Browsing“, macht einen Vorschlag für ein neues Werkzeug, das dem Benutzer zur Verfügung gestellt werden sollte, der zweite, „What you have is what you see“, beschäftigt sich mit der bislang unzureichenden Überschaubarkeit des Materials in Hypersystemen.

Personal Browsing

Hypertexte sind abhängig von der Atomisierung der Informationen des Quelltextes. Ein hoher Grad von Isolation der einzelnen Informationen bedeutet einen hohen Grad von möglichen Verknüpfungen für den Anwender. So wurden bei der Ebstorfer Weltkarte jeder einzelne Textblock, jede einzelne Legende zu einer Grafik als Informationseinheit auf einem eigenen Bildschirm abgelegt.

Diese Zersplitterung bedeutete jedoch auch die Auflösung des komplexen Sinnzusammenhangs der Karte. Die vielfachen Beziehungen der Texte untereinander - z.B. Mythen, die sich in räumlicher Nähe zu bestimmten Orten befinden (wie Babylon: Turmbau, Stadtlegende, Legende zur Region Babylonien) - müssen durch das Programm erst wieder entstehen.

Dies geschieht durch das Anlegen von Verknüpfungen zwischen den einzelnen Texten, die der Benutzer dann später wählen kann. Auch Indexe und Thesauri (Listen von Synonymen wie: Himmel, Firmament) bieten Verknüpfungsmöglichkeiten, die dem Anwender den Stoff strukturieren und ihm neue Assoziationen geben können. Eine Eigenschaft, die eine Volltextsuche nicht leisten kann, da dort nur gesucht werden kann, was bereits bekannt ist. Diese Verknüpfungen erfordern also Interpretationen der Karte, die natürlich immer nur aus der aktuellen Sicht des Programm-Schöpfers und der daran Beteiligten entstehen.

Diese Interpretationen sollen beim EBSKART-Projekt durch „Wanderwege“ realisiert werden, auf denen der Museumsbesucher auf einem vorgegebenen Pfad, unter einem bestimmten Thema durch die Karte geführt wird. So könnte ihn z.B. der Wanderweg „Salzstraße“ durch die historischen Städte und Regionen führen, in denen das Salz im Mittelalter gewonnen, transportiert und abgenommen wurde.

Zur Zeit ist lediglich das Abrufen der einzelnen Textatome möglich, die einen größeren Sinnzusammenhang nur durch das Original an der Wand des Museums, bzw. den Ausschnitt der Bildschirmgrafik erhalten.

Einen wesentlichen Gedanken von Hypertexten sehe ich jedoch darin, den Benutzer von vorgegebenen Pfaden unabhängig zu machen und ihm die Möglichkeit zu geben, eigene Interpretationen zu wagen. Dafür braucht er ein Werkzeug, das ihm ermöglicht, Pfade selbst anzulegen, die seinen Assoziationen und Intentionen entsprechen. Dieses Werkzeug müßte folgende Bedingungen erfüllen:

1. Bleibende Aufzeichnung des Weges, den er gegangen ist.

Dies kann z.B. in Form einer einladbaren Datei geschehen. Damit wäre die Möglichkeit gegeben, mehrere ver-

schiedene Pfade selbst anzulegen und diese unter verschiedenen Anwendern auszutauschen.

2. Der Pfad sollte über eine eigene Navigation verfügen, die auch über das Vor- und Zurücklaufen hinausgeht.

3. Der Pfad muß editierbar sein.

Einzelne Stationen (also Orte auf dem Weg) müssen gelöscht oder eingefügt werden können. Ebenso muß die Reihenfolge veränderbar sein.

4. Der Pfad könnte sinnvoll durch eigene Notizen an beliebiger Stelle erweitert werden. Ebenso wäre eine persönliche Einbindung von Sound oder eigenen Grafiken denkbar.

Damit wäre der Anwender in der Lage, eine eigene Struktur nach seinen Bedürfnissen auf einen vorgegebenen Hypertext zu legen.

What you have is what you see!

Hypermedien erfordern ein neues Nachdenken über das Verhältnis von Form und Inhalt der Informationen. Sie organisieren Datenbestände so, daß sie dem Anwender immer nur einen Ausschnitt des Gesamtdatenbestandes in Form eines Schlüssellochblicks oder nur einen Oberbegriff (Titelzeile, Überschrift) zeigen. Erst durch die Auswahl dieses Oberbegriffs offenbart sich der ganze Text.

Dadurch täuscht sich der Anwender oft über den Umfang und die Art des gesamten Materials. Der 24-bändige Brockhaus enthält mehr Informationen als ein Taschenlexikon von 300 Seiten. Diesen Unterschied kann ich direkt sinnlich an der Menge des bedruckten Papiers wahrnehmen.

Ruft der Benutzer eines Hypertextes eine Informationsmöglichkeit, z.B. eine Titelzeile nicht ab (und er wird im Normalfall immer selektieren müssen), ist er im Unklaren, ob er lediglich einen kleinen Text oder einen sehr großen (vielleicht noch tiefer in Unterpunkte gegliederten) Text überspringt. Überschlage ich ein Kapitel von 50 Seiten in einem Buch oder eines von eineinhalb Seiten Umfang, kann ich die Tragweite dieser Entscheidung besser abschätzen.

Daraus leitet sich die Forderung ab, das dem Anwender gezeigt werden muß, was er hat (was ihm das Programm zur Verfügung stellt).

EBSKART zeigt überwiegend nur Ausschnitte aus der Karte und bei zunehmend größerem Maßstab verschwinden die Details. Jedoch die ständige Anzeige einer Miniaturweltkarte, auf der der aktuell gezeigte Kartenausschnitt invertiert und sich mit dem Maßstab entsprechend verkleinert bzw. vergrößert, gibt dem Anwender ständig die Relation zwischen dem gesamten und dem gezeigten Bestand an.

Auch in Bezug auf die einzelnen Textatome konnte dies befriedigend gelöst werden. Jede Verzweigung auf einen Text ist vorher als Grafik sichtbar. Durch das Anklicken dieses Textbildes wird die Verknüpfung ausgeführt. So signalisiert schon das Bedienelement im voraus den Umfang und m.E. auch den Inhalt des danach aufgerufenen Textes. Z.B. wenn es sich offensichtlich um Städtenamen oder Legenden zu Bildern handelt.

Bei der Anlagerung weiterer Texte wie Sekundärliteratur, Literatur zu einzelnen Mythen oder der Bibel, wird dieses jedoch ein Problem werden. Dann besteht die Gefahr, das der Anwender den Überblick verliert und nicht mehr in der Lage ist, angemessen das zur Verfügung gestellte Material zu nutzen.

Das Mitführen von Zählern (z.B. Karte 4 von 1668) ist nur eine schwache Möglichkeit, da sie zwar die Anzahl der Karten, nicht aber den Umfang der zu erwartenden Texte kennzeichnet.

Die gegenwärtigen Programme gehen noch nicht in Form von Werkzeugen auf dieses Problem ein. Seine Lösung ist noch offen.

Literatur

- [1] Andersen, P.B.: Towards an aesthetics of hypertext systems. A semiotic approach, in: Ritzk, A., Streitz, N. und André, J. (Hrsg.): ECHT '90. The Cambridge Series On Electronic Publishing 224-237. Paris: Cambridge University Press 1990.
- [2] Hahn-Woernle, B.: Die Ebstorfer Weltkarte. Kloster Ebstorf:
- [3] Kugler, H.: Die Ebstorfer Weltkarte. Ein europäisches Weltbild im deutschen Mittelalter. Zeitschrift für deutsches Altertum, Jg. 116 (Heft 1), 1-29 (1987).
- [4] Kugler, H.: Die Texte der Ebstorkarte mit deutscher Übersetzung. 1989. persönl. Mitteilung. Manuskript.
- [5] Warnke, M.: Das Thema ist die ganze Welt: Hypertext im Museum, in: Gloor, P.A. und Streitz, N.A. (Hrsg.): Hypertext und Hypermedia. Informatik-Fachberichte 249, 268-277. Berlin, Heidelberg, New York: Springer-Verlag 1990.
- [6] Warnke, M.: Hypersystem-Projekte im Kulturbetrieb, in: Wallmannsberger, J. (Hrsg.): Hypertext-State of the Art. München: Oldenbourg-Verlag 1991.

KARL CLAUSBERG
Menzelstr. 4
2000 HAMBURG 52

Tel. 040 - 393 443

Lieber Martin,

mein möglicher Beitrag zum *Hypersystem-workshop* würde wohl am besten in die Rubrik *ästhetische Probleme des Medienmix* passen. Ich sehe die ästhetischen Probleme allerdings vor allem auf Seiten einer kognitiven Psychologie. Die hybride Verbindung von Bildern und Texten zum Beispiel hat eine lange Vorgeschichte, aber eine kurze und eigentümlich unsystematische Forschungsgeschichte. Dahinter steckt in der Tat ein fundamentales Problem, das alle Humanwissenschaften berührt: nämlich die Frage, ob es neben den allesdurchsetzenden Sprachstrukturen so etwas wie ein eigenständiges bildliches Denken gibt und in welchem Verhältnis es zur meist als dominant angesehenen Welt der Worte steht. Das Spektrum der Forschungsmeinungen reicht von den perennierenden Ansichten der klassischen Sprachwissenschaft, Herder, Humboldt etc. bis zu den jüngsten Kontroversen um die *mental imagery*, die besonders die englischsprachige Gemeinde erhitzt hat. Viele Argumente in dieser Debatte kommen *pari passu* aus der Gehirn- und Computerforschung. Ich denke, auf diesen flotten Fahrstuhl muß man aufspringen und dort sowohl die Probleme der Akzeptanz, des Daten-*design* etc. als auch einer entsprechenden analytischen Behandlung weiter diskutieren.

Zu diesem Komplex habe ich - als Kunsthistoriker ohne EDV-Phobie - vor dem oben skizzierten Hintergrund viele, viele Fragen; Ich könnte sie entweder in Form von brillianen Diskussionsbeiträgen oder in versammelter Form präsentieren, wenn das ins Konzept paßt.

Schönen Gruß



Gewässerschutz

- ein integriertes Lernprogramm und Informationssystem zum Umweltrecht

I. Einleitung

Umweltrecht einschließlich Gewässerschutzrecht ist in der Ausbildung an Universitäten, Fach- und Verwaltungshochschulen immer noch kein Pflichtfach. Nur an wenigen Universitäten ist das Umweltschutzrecht überhaupt im Curriculum der juristischen Fakultäten vorgesehen. In anderen, insbesondere technischen Studiengängen werden keine juristischen Veranstaltungen zum Umweltschutz angeboten, obwohl die Absolventen in der beruflichen Praxis häufig für die Einhaltung von Umweltschutzvorschriften direkt verantwortlich sind.

Eine empirische Untersuchung des Instituts für Umweltwissenschaften an der Universität Lüneburg ergab, daß in der Wirtschaft ein erhebliches Interesse an Mitarbeiterkenntnissen im Umweltschutzrecht und bei fast 90% an Weiterbildung der Mitarbeiter im Umweltrecht besteht. Dieses Ergebnis wird auch durch die Erfahrung bestätigt, daß die Aus- und Weiterbildungsangebote von Akademien und anderen Bildungsträgern auf große Resonanz stoßen. Auch in der Verwaltung wird ein Mangel an entsprechenden Fachkenntnissen beklagt. Neue Zuständigkeitsverordnungen im Gewässerschutzrecht burden beispielsweise den Gemeinden und Landkreisen die Last der Zuständigkeit für den Gewässerschutz auf, ohne daß die Mitarbeiterinnen und Mitarbeiter in jedem Fall die für die anstehenden Genehmigungsverfahren erforderliche Fachkunde mitbrächten.

Diese hier nur skizzierte Mangelsituation wird durch die Vereinigung Deutschlands und die damit verbundenen Anforderungen an den Umweltschutz in den neuen Bundesländern noch verstärkt.

Der dringliche Bedarf der Praxis an Aus- und Weiterbildung im Umweltrecht war der Anlaß für die Entwicklung eines integrierten elektronischen Lernprogramms und Informationssystems zum Gewässerschutzrecht, das seit Mai 1991 Unternehmen, Behörden sowie Aus- und Weiterbildungseinrichtungen zur Verfügung steht. Die Erfahrungen aus der Evaluation bei zukünftigen Nutzergruppen sind in die marktreife Programmversion eingeflossen. Bei der Evaluierung wurde von der Nutzern positiv hervorgehoben:

- die grafische Unterstützung des Lernstoffs,
- die Möglichkeit, nur bei wirklichem Bedarf Zusatzinformationen aus den Hypertexten zu erhalten,
- das Angebot an sofort erschließbaren Gesetzestexten, die in gängigen Sammlungen nicht enthalten sind,
- die Möglichkeit, auf Tastendruck rechtsvergleichende Untersuchungen der Landesgesetze vornehmen zu können,
- die Möglichkeit, den Grad der "Informationstiefe" frei bestimmen zu können,
- sekundenschnell die relevanten Gesetzestexte auffinden zu können,
- die Freiheit, den Umfang des Lehrstoffs und die Geschwindigkeit des Lernens selbst wählen zu können,
- die einfache, selbst erklärende Bedienbarkeit des Programms, die auch Personen ohne PC-Erfahrung eine sofortige Nutzung ermöglicht.

* Schlütersche Verlagsanstalt Hannover, Unternehmensbereich Elektronische Medien

** Universität Hannover, Institut für Rechtsinformatik, z.Zt. Universität Lüneburg, Institut für Umweltwissenschaften

Das mit einem eigens entwickelten Autorensystem erstellte Programm läuft auf jedem AT-kompatiblen Rechner unter MS-DOS 3.30 aufwärts mit VGA-Grafikkarte und Farbmonitor. Es wird mit Handbuch und Schuber ausgeliefert und nimmt auf der Festplatte ca. 6 MB in Anspruch. Die notwendigen Aktualisierungen der Datenbanken und des Stoffs werden durch regelmäßige Updates sichergestellt.

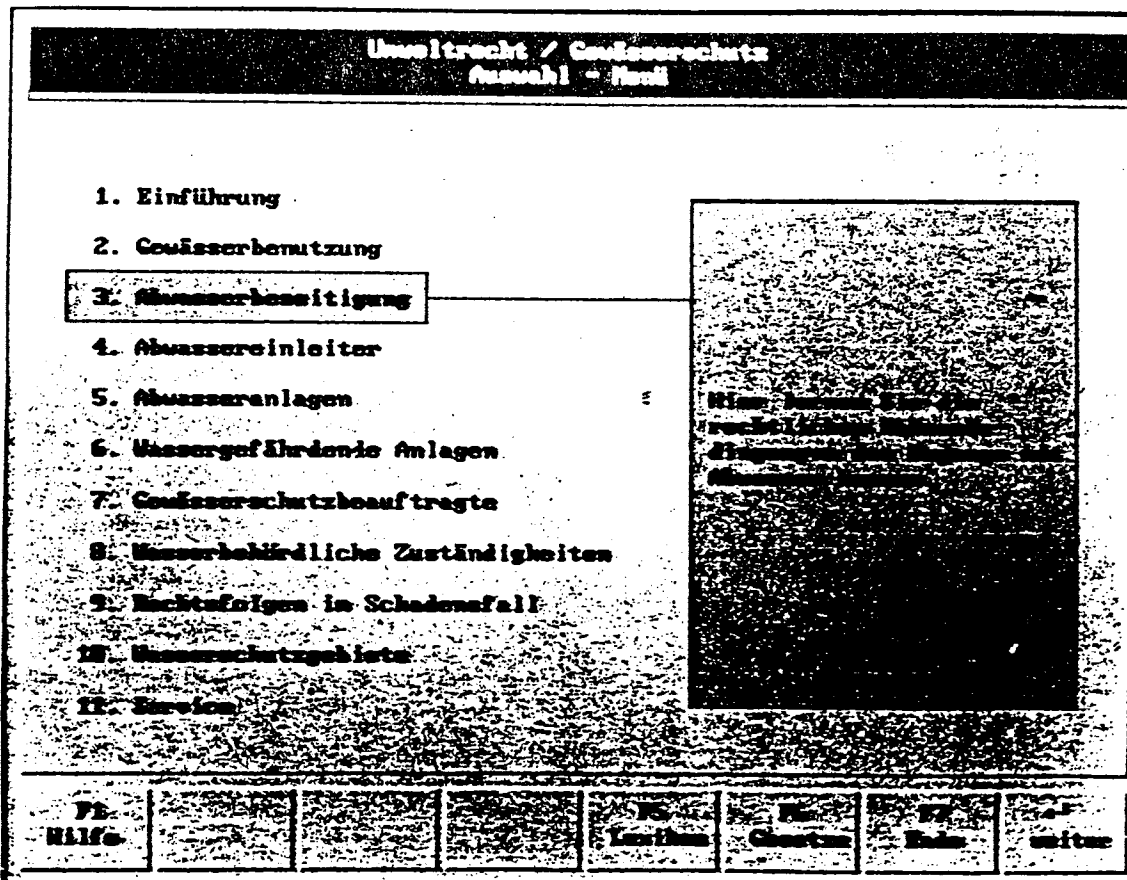
II. Die didaktische Konzeption

Bei der Entwicklung des Gewässerschutzprogramms wurde besonderer Wert auf die Benutzerfreundlichkeit gelegt. Auch in der Benutzung der EDV unerfahrene Personen sollten sofort mit dem Programm arbeiten können. Dementsprechend mußten die Oberfläche und die Bedienung gestaltet sein.

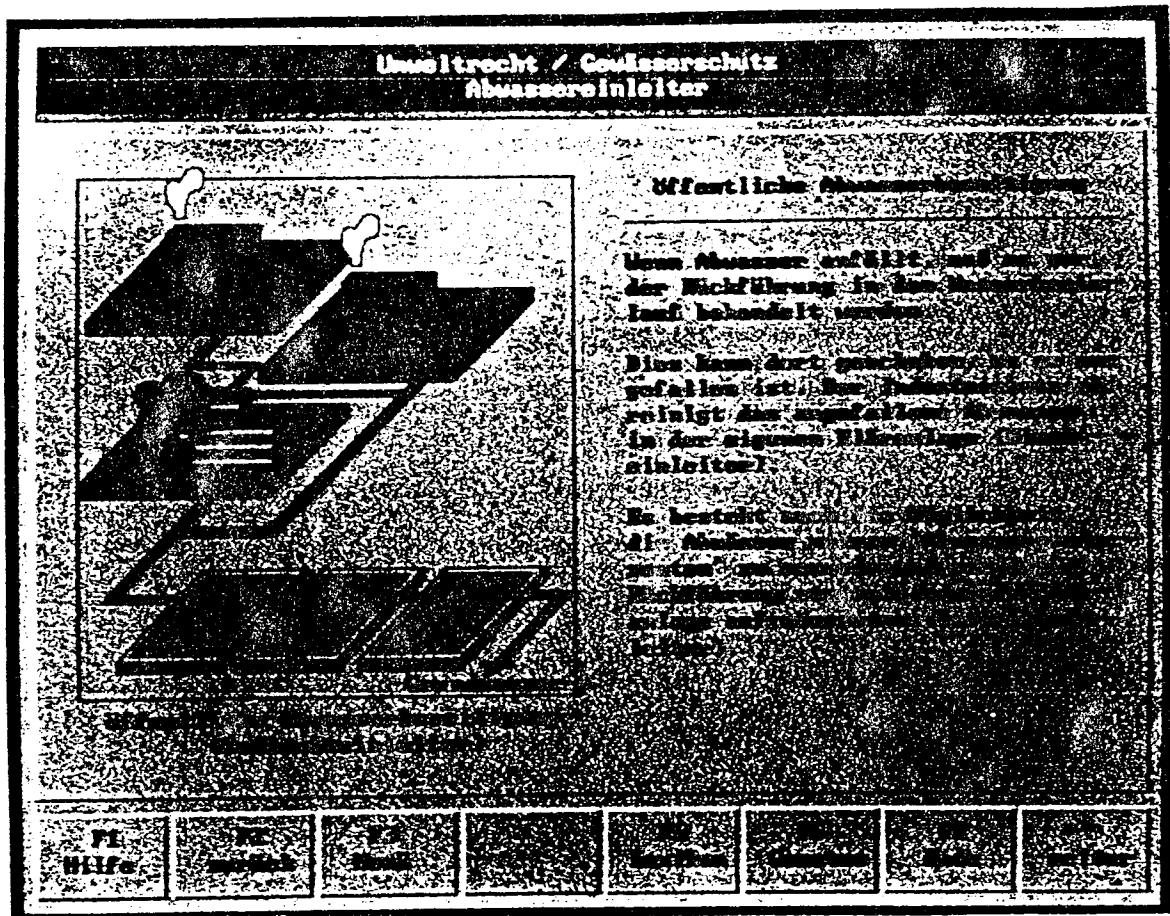
Nach dem Programmaufruf können sich die Nutzer entscheiden, ob sie Informationen über die Bedienung des Programms oder den Programmstart wünschen. Wählen sie die Hinweise über die Bedienung, werden bei Betätigung einer beliebigen Taste ("hit any key") die Bedeutung der Funktions- und Pfeiltasten kurz erklärt. Eine Betätigung anderer als dieser Tasten ist zur Nutzung des Programms nicht erforderlich. Im übrigen kann an jeder Stelle des Programms eine Hilfe mit F1 aufgerufen werden.

Das Programm ist im übrigen so gestaltet, daß der Lernumfang und die Intensität des Lernens den Nutzern überlassen bleibt.

Da das Programm aus mehreren tausend Seiten besteht, muß der Seitenaufbau ausnahmslos gleich bleiben, so daß eine konzentrationsfördernde Gewöhnung und Erwartungshaltung bei dem Aufruf der folgenden Seite eintritt. Auf einer Bildschirmseite dürfen nur die unbedingt notwendigen Orientierungs-, Bedienungs-, und Lerninformationen abgebildet sein.



Die Statuszeile am oberen Bildschirmrand zeigt den Nutzern, an welcher Stelle im Programm sie sich befinden. Das Programm ist hierarchisch aufgebaut. Die Lerninformationen können linear in beliebig wählbaren Teilschritten abgearbeitet werden oder einem bestimmten Informationsbedarf entsprechend aus den Auswahlmenüs gezielt herausgesucht werden. Mit der Funktionstaste F3 kommen die Nutzer sofort zu dem nächsterreichbaren Menü zurück. Sie können sich dadurch schnell im Programm orientieren.



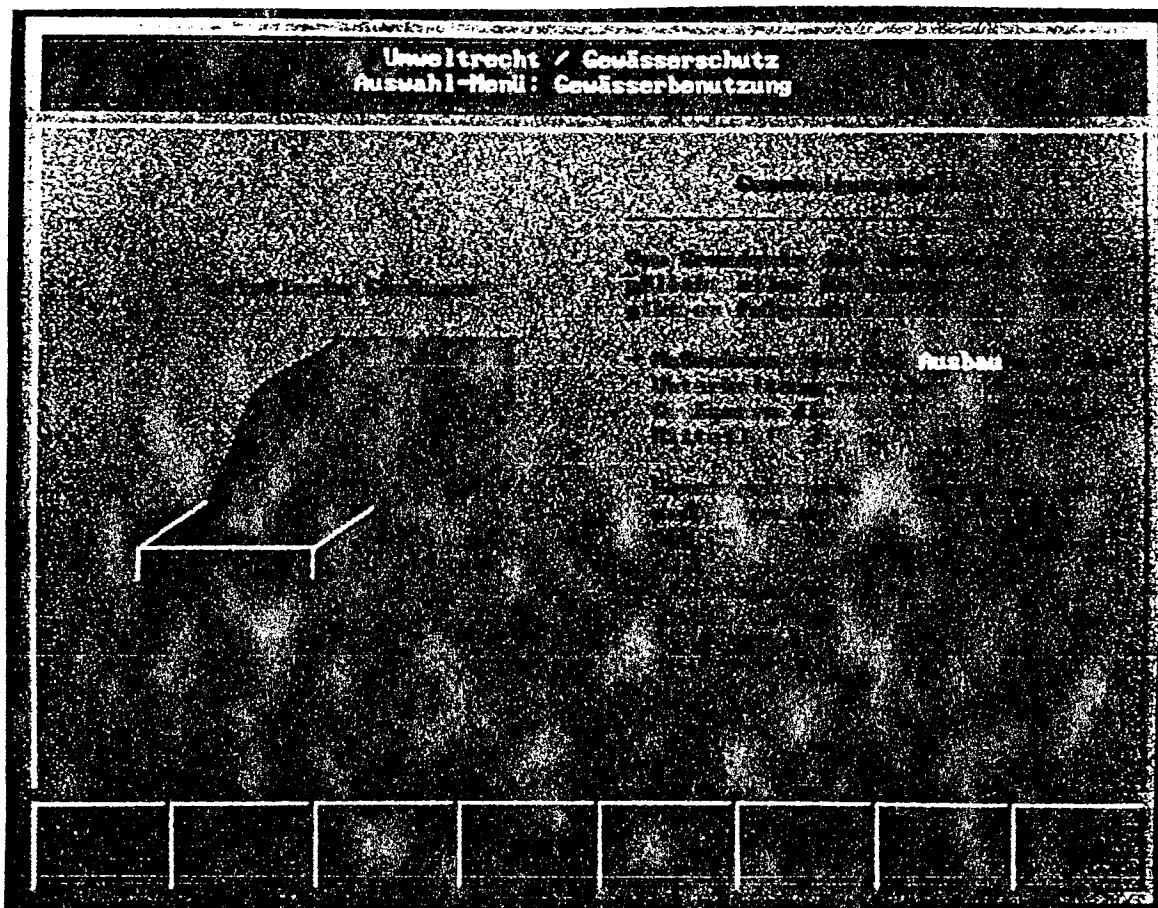
Am unteren Bildschirmrand sind die sieben belegten Funktionstasten erhaben als Tasten dargestellt. Sichtbar sind jeweils nur die Funktionstastenbezeichnungen, die von der angewählten Seite aus aufgerufen werden können.

Der eigentliche Bildschirm selbst wurde in zwei Seiten unterteilt: links finden sich über einhundert mehrfarbige Grafiken, die den konzentrierten Lerntext der rechten Seite veranschaulichen. Die Grafiken dienen der Animation, erläutern komplexe Zusammenhänge optisch und unterstützen das Lernen mnemotechnisch.

Im Hauptmenü können die Kapitel mit den Pfeiltasten angewählt werden, wobei in einem das Textfeld nicht bedeckenden Fenster erste Inhaltshinweise zu dem ausgewählten Kapitel gegeben werden. Ein Aufruf des Kapitels zur näheren Feststellung des Inhalts ist also nicht erforderlich.

III. Das Lernprogramm

Kurze und präzise, in der Diktion und juristischen Terminologie an Nicht-Juristen orientierte Texte führen in das Gewässerschutzrecht ein. Die Nutzer können die Bearbeitungstiefe durch Auswahl der Hypertext-Funktion steigern. Definitionen unbestimmter Rechtsbegriffe, Legaldefinitionen, Leitsätze von Gerichtsentscheidungen oder der Wortlaut von Rechtsvorschriften lassen sich durch Betätigen von F4, immer bezogen auf den jeweiligen Bildschirm, in einem Fenster über der Grafik einblenden, wenn im Text farblich hervorgehobene Begriffe oder Paragraphen die Hypertextanwahl anbieten.



Hypertext wird hier verstanden als zusätzlich aufrufbare, seitenbezogene Information. Das Verlassen des Hypertextes führt unweigerlich wieder auf die Ausgangsseite zurück. Bewußt vermieden wird dadurch ein unkontrolliertes, zufallsgesteuertes, bedingt assoziatives Bearbeiten, wie es bei Hypertext-Systemen im engeren Sinn wie Hypercard zu beobachten ist ("lost in hyperspace").

Den Lernenden wird nach dem Programmstart ein Hauptmenü angeboten, das 10 Lernkapitel und ein Service-Kapitel u.a. mit Hinweisen auf weiterführende Literatur, Anschriften von Organisationen und Verbänden enthält. Die Benutzer können hier auswählen, welchen Wissensstoff sie sich aneignen möchten. Die Auswahlentscheidung wird dadurch unterstützt, daß vor der Auswahl eines Kapitels der wesentliche Inhalt des jeweils gewählten Kapitels kurz erläutert wird.

Nach der Auswahl eines Lernkapitels wird der Wissensstoff kurz vorgestellt. Den Lernenden wird die Möglichkeit geboten, das gesamte Kapitel durchzuarbeiten oder einzelne Unterkapitel aus dem Untermenü auszuwählen.

Inhaltlich gehen die Kapitel auf die folgenden Themen ein:

1. Einführung

Die Einführung würdigt die herausragende Bedeutung des Wassers für das globale Ökosystem. Sie behandelt allgemein das Gewässerschutzrecht und die Rechtsgrundlagen.

2. Gewässerbenutzung

Dieses Kapitel informiert über die gesetzliche Verpflichtung, die Benutzung eines Gewässers genehmigen zu lassen und gibt detaillierte Angaben zu den Genehmigungsvoraussetzungen und -verfahren.

3. Abwasserbeseitigung

Hier lernen die Nutzer die rechtlichen Rahmenbedingungen des Umgangs mit Abwasser kennen und erhalten Auskünfte über schadstoffhaltiges Abwasser und Sonderfälle der Einleitung.

4. Abwassereinleiter

Dieser Programmteil gibt Auskunft darüber, wer Abwasser einleiten darf (Direkt-, Indirekteinleiter) und welche Abgaben für Einleitungen zu entrichten sind.

5. Abwasseranlagen

In diesem Kapitel werden die Anlagen vorgestellt, die der Behandlung des Abwassers dienen.

6. Wassergefährdende Anlagen

Hier werden die wichtigsten Regelungen über Rohrleitungsanlagen und über Anlagen zum Umgang mit wassergefährdenden Stoffen vorgestellt.

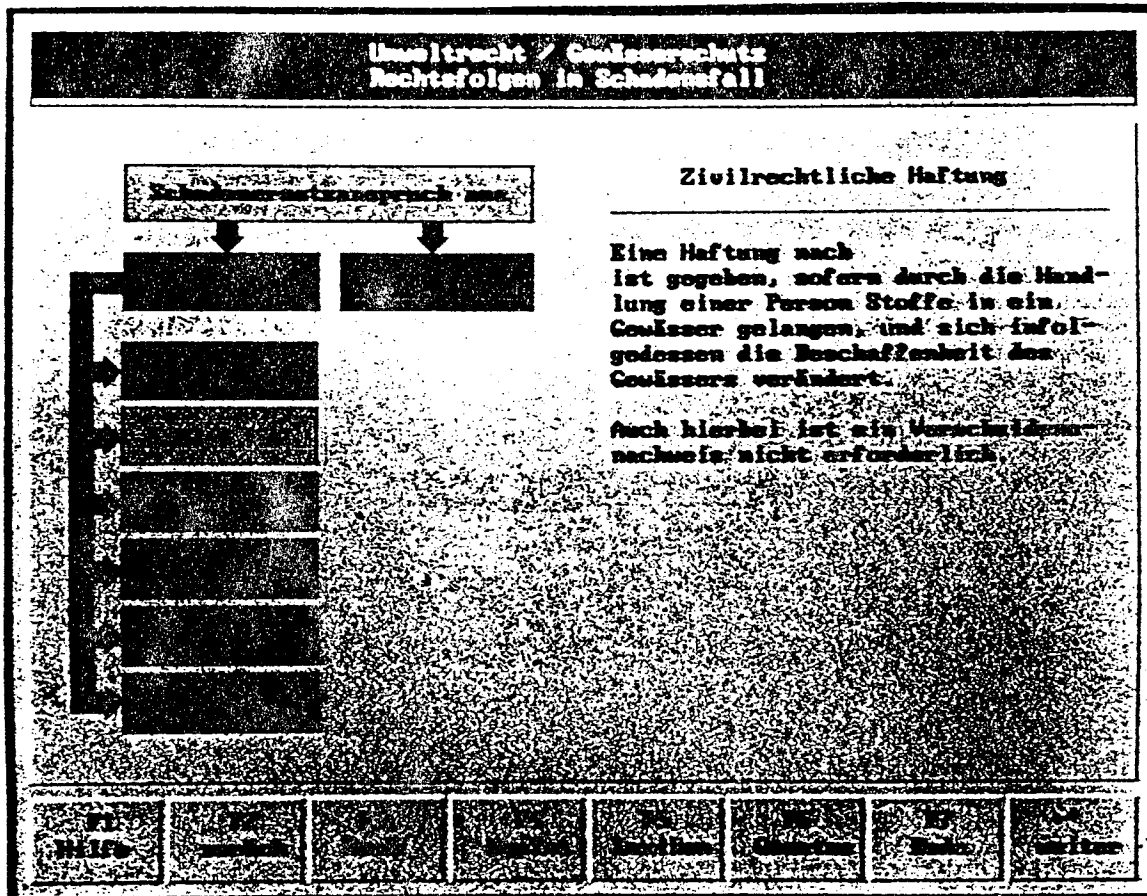
7. Gewässerschutzbeauftragter

Es werden die Einrichtung des Gewässerschutzbeauftragten, dessen Aufgaben und die Pflichten des Gewässerbenutzer behandelt

Umweltrecht / Gewässerschutz Der Gewässerschutzbeauftragte	
	Aufgaben des Gewässerschutzbeauftragten
	Ziel dieses Kapitels ist es, die Aufgaben und Pflichten des Gewässerschutzbeauftragten zu erläutern.
	Abwasser
Überwachungsaufgabe (§ 21b Abs. 1 Nr. 1 WHG)	
Initiativsaufgabe (§ 21b Abs. 1 Nr. 2 und 3 WHG)	
Informationsaufgabe (§ 21b Abs. 1 Nr. 4 WHG)	
Berichtspflicht (§ 21b Abs 2 WHG)	

8. Rechtsfolgen im Schadensfall

Dieses Kapitel stellt die strafrechtlichen Sanktionen und die zivilrechtlichen Haftungsfolgen bei einer Gewässerbeeinträchtigung dar.



9. Wasserschutzgebiete

Hier wird beschrieben, auf welchen rechtlichen Grundlagen die Ausweisung von Wasserschutzgebieten beruht.

10. Wasserbehördliche Zuständigkeiten

Differenziert nach dem jeweiligen Bundesland werden die behördlichen Zuständigkeiten erläutert.

III. Das Informationsprogramm

Das Programm Gewässerschutzrecht wurde nicht nur als Lernprogramm, sondern auch als Informationssystem konzipiert. Ein Informationssystem ist es aus folgenden Gründen:

Nutzer, die spezielle Informationen erhalten wollen, können diese gezielt aus dem Programm abfragen. So kann ein Unternehmen sich beispielsweise informieren, ob eine bestimmte Gewässerbenutzung genehmigungsbedürftig ist, ob mit Auflagen oder Bedingungen zu rechnen ist, wegen dieser Gewässerbenutzung ein Gewässerschutzbeauftragter bestellt werden muß, welche Behörde die zuständige Genehmigungsbehörde ist, welche Anschrift sie hat und ob von Seiten der EG, des Bundes oder des Bundeslandes Förderprogramme vorgesehen sind, die Anreize für den Einbau einer Abwasserreinigungsanlage bieten.

Unmittelbar Informationsfunktion hat das "Service"-Kapitel, das


- eine komplette Übersicht über die Förderprogramme auf dem Gebiet des Gewässerschutzes für die gewerbliche Wirtschaft,
- Anschriften von staatlichen und nichtstaatlichen Stellen des Bundes und der 16 Länder im Bereich Umweltschutz und Wasserwirtschaft,
- eine umfangreiche Literaturübersicht und
- den gesamten Katalog aller definierten wassergefährdenden Stoffe und ihrer Wassergefährdungsklasse

enthält.

Von jeder Stelle im Programm sind die Datenbanken "Lexikon" und "Gesetze" anwählbar.

Umweltrecht / Gewässerschutz
 Wasserbehördliche Zuständigkeiten

Brandenburg



Nach dem Beitritt gilt aufgrund Anlage II, Kapitel XII Abschnitt III des Einigungsvertrags vom 23.9.1990 (BGBl. II Nr. 35 vom 28.9.1990) das Umweltrahmengesetz der DDR vom 29.6.1990 fort.

Mit dem Umweltrahmengesetz wurden für das Gebiet der ehemaligen DDR das Wasserhaushaltsgesetz, die Abwasserbeseitigungsverordnung und Verwaltungsvorschriften nach § 7a WHG, das Abwasserabgabengesetz und das Wasch- und Reinigungsmittelgesetz übernommen.

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16.

Das Lexikon enthält ein alphabetisch geordnetes Glossar mit den relevanten Fachbegriffen aus dem Gewässerschutzrecht. In der Datenbank Gesetze, deren Nutzung keine Retrieval-Kenntnisse erfordert, sind weit über dreitausend Seiten Gesetzes- und Verordnungstexte sowie Verwaltungsvorschriften der Europäischen Gemeinschaften, der Bundesrepublik Deutschland und der Bundesländer enthalten und über ein Menüsystem schnell erschließbar. Eine vergleichbare elektronische Rechtsdatenbank zum Gewässerschutzrecht, die auch die Landesvorschriften enthält, gibt es in Deutschland nicht. Wenn das Lexikon oder die Gesetzesdatenbank geschlossen wird, finden sich die Nutzer wieder an der Programmstelle, von der aus sie die Zusatzinformationen abgefragt hatten.

IV. Zusammenfassung

Das CBT-Programm Umweltrecht - Gewässerschutz ist ein interaktives Informationssystem und Lernprogramm für Wirtschaft, Verwaltung und Ausbildung mit grafisch unterstützter Benutzerführung, mehrschichtiger Parallelstruktur, seitenbezogener Recherche (Hypertext) und Datenbanken mit Direkteinstieg.

Anwendung von Hypersystemen in der Medizin

Bei stetig zunehmender Informationsmenge und Komplexität sowie gleichzeitig abnehmender "Halbwertszeit" von Informationen bieten sich Computerlösungen zur Aufbereitung und Darstellung dieses Wissens an.

Ein Wissenselement (z.B. die Aortenstenose) läßt sich dabei im Hypersystem aus unterschiedlichen, aber doch zusammengehörigen, Richtungen ("Achsen") betrachten. Zu diesen Achsen gehören Text, Bild, Diagramm, formelhafter Zusammenhang, Audio, Animation und andere inhaltlich verwandte Informationsquellen. Die verschiedenen Informations-Achsen ergeben in ihrer Gesamtheit das Wissenselement.

Wir verstehen Hypermedia als Hilfsmittel, welches diese Achsen darstellt, verknüpft und für den Benutzer in einfacher Weise bearbeitbar macht.

Für den Einsatz in der Medizin konzipieren und realisieren wir Informations- und Lernsysteme.

Projekt 1: Ein medizinisches Lexikon.

Bei der Entwicklung dieses Lexikons sind wir von bestehenden Büchern (Pschyrembel, Roche Lexikon Medizin) ausgegangen. Zusätzlich wurden Informationen, die über den Umfang des Buches hinausgehen eingebunden.

In der Konzeptphase haben wir die Elemente zur Informationsvermittlung definiert. Es sind dies Text, Abbildungen, Diagramme, Audio und Formelsammlungen. Die Kombination dieser Elemente erfolgte unter den Gesichtspunkten des Informationsgewinns und der intuitiven Eingängigkeit.

Als Kriterien für die Entwicklung der Funktionalität des Systems haben wir herausgearbeitet:

- integriertes Tutorial zur Einarbeitung in die Bedienung
- Übersichtlichkeit
- eingängiges, offensichtliches und konsistentes Bedienungsverhalten des Systems
- On line Help-Funktion
- kein Hyperlink ohne begleitende Navigatorfunktion ("wo bin ich?")
- Möglichkeit der Eingabe von Bemerkungen (Notizfunktion)
- vom Benutzer frei wählbare und speicherbare Zusammenstellungen von Informationen zu sog. Themen

Die Entwicklung des Konzepts wird in erster Linie aus der Sicht der Anwender (Ärzte, Studenten und medizinische Assistenzberufe) und deren Informationsbedürfnissen und Anforderungen an die Bedienung vorangetrieben.

Das Lexikon wurde mit Hypercard erstellt.

Im Verlauf des Projektes haben sich bei der Umsetzung des Wissens aus der klassischen Buchform in die des Hypersystems eine Fülle von Problemen ergeben. Beispiele hierfür sind der gesamte Komplex der Bedienerführung, die Frage der Oberflächengestaltung und der Visualisierung von Funktionszuständen des Systems sowie die Navigation "entlang" der Hyperlinks.

Wir wollen unsere Lösungsansätze zur Diskussion stellen und hoffen, bei manchen der offenen Fragen gemeinsam weiter zukommen.



Projekt 2: Ein Lernprogramm.

Wir arbeiten an einem Lernprogramm zu "Grundlagen der Mikrobiologie und der antiinfektiösen Therapie". Neben den Elementen des Lexikons (Text, Abbildungen, Diagramme und Audio) wurden vermehrt multimediale Aspekte berücksichtigt. Diese betreffen vor allem die Simulation und die Veranschaulichung von Vorgängen, die sich in zeitlicher oder in räumlicher Dimension verändern. In der Konzeptarbeit haben wir versucht, die bestehenden und erfolgreichen Ansätze der Wissensvermittlung (Vortrag, Lehrbuch, programmiertes Lernen und Telekolleg) zu analysieren und -wo sinnvoll- zu adaptieren.

Zu den oben genannten Kriterien des Lexikons sind für die Funktionalität des Lernprogrammes zusätzlich folgende Punkte von Bedeutung:

- zeitliche Ablauf des Programmes (anpaßbar an das jeweilige Lernverhalten)
- Selektion und Gliederung des Stoffes
- Balance zwischen den einzelnen multimedialen Elementen
- Synergismus und Synchronisation der einzelnen multimedialen Elemente
- Benutzerführung im zeitlichen und inhaltlichen Ablauf

Diese Lernsoftware wird mit MacroMind Director, Swivel 3-D und HyperCard verwirklicht. Im Zusammenspiel dieser Elemente liegt eine besondere Herausforderung. Neben dieser sehen wir auch in der Frage der "lernwirksamen" Aufbereitung und Darbietung des Stoffes Grundlagen für eine interessante Diskussion.

Neben den Vorträgen zu den oben genannten Projekten können beide Programme auch demonstriert werden.

Norbert Bolz / Berlin

**Navigation im docuverse. Perspektiven einer
hypermedialen Darstellung von Komplexität**

Seit sich die Kinder an die Multi-Media-Environments der großen Städte und die unendlichen Datenflüsse der neuen Medien verlieren, haben die Bildungsstrategien der Gutenberg-Galaxis ausgespielt. Swamped by information overload sitzt schon das Vorschulkind als Jäger vor Bildschirmen. Alles Suchen und Forschen läuft über pattern recognition.

Rückblickend kann man Marshall McLuhans Medientheorie und sogar Walter Benjamins Passagenwerk als Versuche begreifen, dieser neuen Medienwelt darstellungstechnisch gerecht zu werden. Es sind keine Bücher mehr, sondern Mosaike aus Zitaten und Gedankensplittern - Schreiben wie Kino. Diese Versuche, Bücher zu schreiben, die die Buchform sprengen, sind gerade in ihrem Scheitern lehrreich. Offensichtlich ist das Informationsverarbeitungssystem Buch der Komplexität unserer sozialen Systeme nicht mehr gewachsen. Deshalb organisieren Autoren, die das wissen und doch Autoren bleiben wollen, ihre Bücher nach Strukturen und Mustern, die nicht-linearen Informationsverarbeitungssystemen entwendet sind. Prominentestes Beispiel ist natürlich Hofstadters eternal golden braid. Das Buch argumentiert mit selbstverstärkenden Resonanzen, die durch strange loops zwischen verschiedenen Ebenen der Reflexion/Darstellung entstehen. Was man früher als infiniten Regress perhorresziert hätte, wird elegant in eine wissenschaftliche Ästhetik der Rekursion umgedeutet. Doch auch ein Buch über strange loops das wie ein strange loop konstruiert ist, bleibt ein Buch. Gerade weil es nichtlinear aufgebaut ist, wird sich, wer es von Deckel zu Deckel liest, unweigerlich langweilen.

Labyrinthische Argumentationen - von Luhmann bis Hofstadter - lassen sich eben nicht in Buchform sequenzieren. Um die hier angezielte Komplexität ohne Informationsverlust darzustellen, wäre eben eine Simultanpräsentation in mehreren Ebenen nötig. Niklas Luhmann, der seine Daten ja immer noch mit Zettelkästen verwaltet, hat das sehr klar gesehen. Darstellungen von Komplexität erfordern komplexe Theorien, die nicht mehr serialisierbar sind, weil die

Reflexion an mehreren Einstiegspunkten zugleich ansetzt. Auch präsupponiert jeder Reflexionseinstieg eine Vielzahl nichtexplizierbarer Voraussetzungen. All das kann in linearer Lektüre nicht realisiert werden. Gesucht wird also ein Medium simultanpräsentere Darstellung: "Schön wäre es, wenn man diese leicht labyrinthische Theorieanlage in Büchern abbilden könnte, die sozusagen zweidimensional angelegt sind, also mehrere Lesewege eröffnen." (Luhmann)

Eben das aber leisten Hypermedien. Sie haben eine starke Affinität zu den RDBMS, den relational database management systems, die dynamische Strukturen mit relationalen Operatoren abarbeiten; d.h. sie arbeiten - weit entfernt vom linearen Datenprocessing - mit many-to-many-Relationen. Und hier zeichnet sich nun eine für jene komplexe Theorie des Komplexen entscheidende Implementierungsmöglichkeit ab: Elaborierte Hypermedien werden mit second-order-links operieren, d.h. also mit links to links. Damit wäre ihre software-Struktur aber identisch mit der Theorie-Struktur komplexer Sachverhalte: Relationierung von Relationen.

Auch die jetzt schon jedem Benutzer von Hypermedien vertrauten typed links sind tools fürs Komplexitätsmanagement. Sie reduzieren die Datenkomplexität, indem sie eine Vielfalt virtueller Navigationsperspektiven eröffnen. Das darf nicht mit herkömmlichen Klassifikationsverfahren verwechselt werden. Hypermedien kennen keine lingua franca der Darstellung von Wissen. Sie integrieren heterogenstes Material, operieren auch fuzzy und probabilistisch, prozessieren parallel und "konnektionistisch" und können deshalb ideale brainstorming tools bereitstellen.

Der Informationsraum der Hypermedienkultur hat n Dimensionen. Und gerade weil Information keine natürliche Topographie hat, intervenieren auf der Benutzeroberfläche immer wieder orientierende Medienmetaphern: Bibliothek, Kompaß, Netz, Film, Hitch Hiker's Guide. Hypermedien benutzen also die charakteristischen Leistungen älterer Medien als metaphorische Navigationshilfen im n -dimensionalen Informationsraum. Vor allem die Filmmetaphern etwa auf Apples HyperCard markieren den Abschied von den linearen Informationsstrukturen: Auflösung, Schnitt, Montage, flashback, zoom in/out. Solche nichtlinearen Strukturen des Datenmanagements erzeugen natürlich eine viel höhere Verknüpfungsdichte (busy screen), als dies Büchern je möglich wäre. Darauf reagieren Bücher recht hilflos mit den schon erwähnten strange-loop- und Labyrinth-Techniken, mit Spezialzeichen, die Aufmerksamkeit steuern und

Textstufen indizieren sollen (z.B. in Mandelbrots Fraktaler Geometrie der Natur) - oder eben damit, daß Disketten eingeklebt werden. Mit solchen hilflosen Gesten tastet die Gutenberg-Galaxis nach den Möglichkeiten des docuverse. Erst Hypermedien ermöglichen aber, was bisher nicht implementierbar war: das browsing between media. Es wird ein nahtloses Hypermedia-Environment konstituieren, das sich durch interaktiven Informationsgebrauch reproduziert. Ginge es allein nach der Logik der Hypersysteme, so würde die neue Textgesellschaft - Ted Nelson hat sie docuverse genannt - entschlossen Abschied nehmen von privaten, diskreten Dokumenten und sich in der "freien Rede" unendlicher (Re-)Kombinationen von fragment showers of information entfalten. Doch auf absehbare Zeit wird es schon deshalb immer noch "Autoren" geben, weil die kapitalistische Spezifikation des Projekts Textgesellschaft auf dem copyright bestehen muß. Piraterie wird zum Schlüsselproblem. Deshalb - und nicht etwa aus technischen Gründen - müssen etwa CD-ROMs bleiben, was sie heißen: eben read only memories. Die Frage nach Autorschaft, copyright und Piraterie markiert den entscheidenden Verzweigungspunkt zwischen (noch einmal Ted Nelson:) closed und compound hypermedia.

Workshop

»Hypersystem- Konzepte in Medien und kultureller Produktion«

15. bis 17. Juli 1991 im Rechen-
zentrum der Universität Lüneburg,
Stresemanstr. 6 in 2120 Lüneburg

Programm und Anmeldung

Fachbereich 8 »Informatik und
Gesellschaft«
und
Fachgruppe 4.9.1 »Hypertextsysteme«
der Gesellschaft für Informatik

Das Thema des Workshops

Hypertext- und Multimedia-Systeme sind technisch zum Teil recht weit entwickelt, inhaltliche Fragestellungen jedoch, wie die nach einer Ästhetik der Hypersysteme, nach Produktion und Rezeption hypermedialer Produkte sowie ihres Verhältnisses zu anderen Medien bedürfen noch weitgehend der Klärung.

Der Workshop »Hypersystem-Konzepte in Medien und kultureller Produktion« hat für die Bearbeitung dieser Probleme folgende Schwerpunkte gesetzt:

Ästhetik der Hypersysteme
Produktion und Rezeption von Hypersystemen
Expertensysteme und Hypersysteme sowie
Lernsysteme.

Die Themen werden in Form von Vorträgen, Systemdemonstrationen und Produktvorstellungen (unter denen auch ein interaktiver Film vertreten ist) bearbeitet.

Anmeldung zur Teilnahme

Bitte senden Sie das beiliegende Formular ausgefüllt bis zum 15. 6. 1991 zurück.

Die Teilnahme ist kostenfrei.

Die Teilnehmerzahl ist auf 50 beschränkt. Über die Teilnahme entscheidet die Reihenfolge, in der die Anmeldungen eingehen.

Unterbringung in Lüneburg

Ihre Unterbringung in Lüneburg müssen Sie selbst organisieren. Eine Liste der Unterbringungsmöglichkeiten in Lüneburg und Umgebung liegt bei.

Nachfragen zur Veranstaltung richten Sie bitte an:

Dr. Martin Warnke
Universität Lüneburg
Stresemanstr. 6
D-2120 Lüneburg
Tel.: (04131) 714 471
Fax: (04131) 714 443

Ausstellung

Im Rahmen einer Ausstellung können viele der Systeme, die im wissenschaftlichen Programm vorgestellt werden, in näherem Augenschein genommen werden. Außerdem sind Hard- und Software zu sehen, die besondere Bedeutung für die Produktion multimedialer Dokumente haben. Die Liste der Ausstellungsstücke lautet:

„HYDRA“

Ein Museums-Auskunfts-System

„Ein medizinisches Lexikon“ und Lernprogramm „Grundlagen der Mikrobiologie und der antinfektiösen Therapie“

„Archivierung und Entzifferung der ältesten Texte der Menschheit“

„Schwamm“

„Lernen mit Hypertext“

„Die Imaginäre Bibliothek“

„Palenque“

„HyperXPer“

„Lernprogramm und Informationssystem „Gewässerschutzrecht“

„EbsKart“ und Videomaterial zu „EbsKart“

„HyFi“

Filmmaterial zu „Performance-Hyperkatalogen“

Digital Video von SuperVac, Dia- und Mikrofilmscanner, Dokumenten-Verwaltungs-System von Archis

Montag, 15.7.91

9:30-10:00 Begrüßung

durch den Rektor der Universität Lüneburg, Herrn Prof. Dr. Hartwig Donner und Eröffnung des Workshops durch Martin Warnke

10:00-11:00 Ästhetik der Hypersysteme I

Heiko Idensen, Matthias Krohn: „Entwurf einer Hyperpoetik“

Ausstellung/Kaffee

11:30-12:30 Ästhetik der Hypersysteme II

Peter Bøgh Andersen, Berit Holmqvist: „On the aesthetics and rhetorics of interactive narrative systems: Interaction as sign production“

Ausstellung/Mittag

14:00-16:00 Produktion und Rezeption von Hypersystemen I

Frank Voeltz, Thomas Tesch, Tom Quellenberg, Ulf Hönisch: „Das Arbeitsvorhaben HyFi: ein hypermedialer Filmkatalog“

Rita Abel, Jörg Blankemeyer, Wolfgang Coy, Jörg Geißler, Ferruh Mercan, Ingo R. Meier, Oliver Schoorman, Antje Tietjen, Torsten Rahn: „HYDRA – Filmanalyse im rechnergestützten Medienverbund“

Ausstellung/Kaffee

16:30-18:30 Ästhetik der Hypersysteme III

Peter Gendolla: „Text-Tänze, Anagramme und Adaptionen: Vorbereitungen für eine Ästhetik der Hypersysteme bei R. Rousset, H. Bellmer und O. Wiener“

Kurd Alsleben (Interdisziplinäre Computerei der HBK Hamburg): „Dialogische Künstlerrolle (von HandlochKart' zu Hai'pe:kat:d)“

18:30 Empfang

Dienstag, 16.7.91

9:00-11:00 Expertensysteme und Hypersysteme

Fahri Yetim: „Integration von Expertensystem- und Hypertext-Techniken am Beispiel des Systems WISKREDAS“

Annette Meini: „HyperXPerit – Hypertext und Expertensysteme“

Ausstellung/Kaffee

11:30-12:30 Lernsysteme I

Ulrich Glowalla: „Lernen mit Hypertext“

Ausstellung/Mittag

14:00-16:00 Produktion und Rezeption von Hypersystemen II

Peter Damerow, Martin Schreiber: „Hypertext und Datenbank-Technik am Wissenschaftler-Arbeitsplatz“

Paul Ferdinand Sieger, Walter Uka, Martin Warnke: „Das Projekt EbsKart – Medienmix in einem interaktiven Auskunfts-system“

Ausstellung/Kaffee

16:30-17:30 Ästhetik der Hypersysteme IV

Karl Clausberg: „Medienmix – Im Kopf und vor Augen“

18:00-20:00 Ein interaktiver Film:

Peter Krieg: „Suspicious Minds“ (Die Ordnung des Chaos)

Mittwoch, 17.7.91

9:00-11:00 Lernsysteme II

Erich Röhrich, Jürgen Taeger: „Lernprogramm und Informationssystem »Gewässerschutzrecht“

Ingo Carl, Aly Sabri: „Ein medizinisches Lexikon“ und Lernprogramm „Grundlagen der Mikrobiologie und der antiinfektiösen Therapie“

Ausstellung/Kaffee

11:30-13:30 Ästhetik der Hypersysteme V

Thomas Hölischer: „Der Bildbegriff in der Kunstgeschichte und in der Computer Vision sowie den elektronischen Bildmedien“

Norbert Bolz: „Navigation im docuverse. Perspektiven einer hypermedialen Darstellung von Komplexität.“

Ausstellung/Mittag

15:00 Abschlußsitzung

„Hypersysteme: wohin soll die Welle schwappen? Perspektiven, Kooperationen, Organisationen“