From Ideas and Arguments to Hyperdocuments: Travelling through Activity Spaces

Norbert A. Streitz, Jörg Hannemann, and Manfred Thüring

Integrated Publication and Information Systems Institute (IPSI)
Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD)
Dolivostraße 15
D-6100 Darmstadt
F. R. Germany (F.R. Germany)

INTRODUCTION

Discussing relevant issues for the next generation of hypermedia systems, Halasz [Hala88] provides also a classification along the following three dimensions: scope, browsing vs. authoring, and target task domain. In this paper, we will especially discuss aspects of the second dimension focussing on support for idea processing and authoring in hypertext systems. Although one cannot classify existing systems by assigning them exclusively to one category of this dimension[1] hypertext systems are primarily discussed from the reading and browsing point of view and as support for retrieval. This is also reflected in attempts to transform existing (linear) text sources into hypertext structures in order to profit from their additional interactive branching capabilities.

On the other hand, if one really wants to make use of the full concept of hypertext structures as described by Conklin [Conk87], this offers new and exciting possibilities for writing and for ways to support this activity. From our point of view [Stre88b], writing is a complex problem solving and design activity with multiple constraints. The final product - in terms of a hyperdocument - can be viewed as an externalized representation of internal knowledge structures which have been developed by the author. Thus, authoring tools which are especially geared to the preparation of hyperdocuments will offer much better facilities for conveying the message and intention of authors. This way, they can communicate knowledge in a format which is closer to their knowledge structures than it was possible with traditional documents. Integrating additional information about the author's intentions and knowledge structure and shipping it to the reader as part of an electronic document facilitates more comprehensive processing on the recipient's side. It implies that documents produced with these tools keep authors' knowledge structures alive by preserving their argumentation and rhetorical structures which then can be used for subsequent processing. This improves not only reception by human readers but also by text analysis components for machine translation or automated abstracting. While it is very difficult today to analyse argumentative and rhetorical aspects of natural language texts, these documents would contain this information explicitly. We will provide examples of this additional information when discussing the activity and document type "argumentation" which we selected as our task domain.

Another starting point for our research is the observation that almost all hypertext systems are - especially with respect to authoring - passive systems, i.e. they do not offer active (intelligent) support to the author by providing feedback, advice, or guiding. Of

[1] A clearcut distinction has been employed only with the separation of Concordia and the Document Examiner [Walk87].

course, the realization of this goal requires the integration of knowledge-based capabilities in a hypertext system. This implies that the architecture of the authoring tool includes components which permit monitoring and analysis of the author's activities. They are necessary to build up knowledge bases about authors and semantic structures of documents. But these provisions are rarely found in existing systems.

Based on this assessment of the current situation of hypertext systems, we are developing an active, knowledge-based authoring and idea processing tool for creating and revising hyperdocuments. The system - SEPIA: Structured Elicitation and Processing of Ideas for Authoring - will represent a major portion of the functionality we expect from an author's workbench of the future. The research presented in this paper represents only a part of it and must be viewed within this framework.

System design and prototype development of SEPIA is determined by two main objectives. Our first and overall objective is to build a cognitively adequate system. To accomplish this, we employ the approach of user-oriented and task-driven system design based on the principle of cognitive compatibility [Stre87]. This requires to use results derived from models about cognitive processes in writing and more general in problem solving. The second more ambitious objective is to build an active system. Especially by setting our focus this way, it was necessary to cut down the complexity of the general problem of active systems. We selected a finite but not trivial subset of authoring activities and document types: argumentation and argumentative texts. The activity as well as the document type are very well defined and exhibit structures which offer excellent starting points for monitoring and guiding.

AUTHORING SYSTEMS AND WRITING IN HYPERTEXT ENVIRONMENTS

As already mentioned in the introduction, hypertext systems can be distinguished by their primarily intended use: retrieving (browsing and searching) vs. authoring (creating and designing). This distinction does not exclude that some systems can be used for both purposes, e.g. KMS [Aksc88] or Intermedia [Yank85], but most of them differ in the extent of support for these two activities. Examples of the first type are HyperTIES [Marc88] with its usage in interactive museum exihibits and the Document Examiner [Walk87] as the on-line presentation of technical documentation from Symbolics. Although it also allows to create and organize personally structured selections of the total information offered it is a read-only system. The corresponding authoring tool (Concordia) has been made available separately.

Systems primarily designed for writing focus on tools for creating nodes and links, organizing them into network structures and revising these structures as well as the content of their nodes. Examples of authoring tools are: TEXTNET [Trig86] and NoteCards [Hala87] for idea processing, Writing Environment [Smit87] for document preparation or Neptune [Deli86] for supporting the design and documentation of large-scale software systems. Since Neptune is meant to support software engineering it also stresses versioning and node/link attributes. Systems as e.g. Guide [Brow87], HyperCard [Will87] and HyperTIES also show some limitations as they do not offer a graphical representation of the node-link structure of the hypertext network. This deficit results in a number of crucial problems connected with support for navigation (disorientation problem) and for personal information structuring. A special but for our intended research relevant application is to map argumentation structures onto hypertext structures. One of the rare examples is the gIBIS-system [Conk87b], a hypertext tool for team design deliberation.

Two general problems for authoring in hypertext environments can be identified. First, the problem of "cognitive overhead" [Conk87a] which results from the requirement to label nodes, links and structural relationships at a very early stage. NoteCards [Hala88] forces the user to label and link his nodes right away which often results in premature organization. This conflicts with a "natural" way of generating ideas and writing initial segments of text when structures are less definite in the beginning. Most of the time, they are not explicitly spelled out, exist only in the mind and evolve in a flexible way much later. Second, almost all authoring tools for hypertext are rather passive storage and retrieval systems, i.e. they do not provide active support compatible with the activities of the authors. By "active" we mean that the authoring tool should be able to monitor and guide authors in their problem solving activity.

In general, one can state that the issue of providing cognitively adequate support for authors of hypertext documents has not been addressed as much as is desirable. One exception is the approach connected with the development of the Writing Environment (WE) [Smit87, Smit88]. In this case, design decisions were based within a cognitive framework of writing. This approach is a very promising one, but the specific implementation is lacking some features of the full hypertext concept. Although WE provides a linking mechanism in the network mode the resulting structure is not preserved in the final document. Beyond this, there are more basic deficits in hypertext research with respect to writing. Research is not really addressing the crucial problem that writing a non-linear text might require very different concepts of creating, revising, and composing documents and therefore different kinds of support. On the other hand, the publishing situation to date does not really provide the external conditions and demands for having hypertext as the final document structure. This is also reflected in reports about authors using e.g. NoteCards for idea processing and structuring but turning to outliners and traditional text processing tools when writing the final document [Trig87].

DEVELOPMENT OF AUTHORING TOOLS AND THEORIES OF WRITING

A prerequisite for information retrieval - not only in hypertext systems - is that this information must have been produced some time before. While publishing is communicating knowledge, authoring and in particular writing is knowledge production and transformation. It can be observed that the construction of writing tools is mainly based on intuition and first-order task analysis. What is lacking is a sound theoretical foundation for building cognitively compatible interfaces which provide intelligent support for writing. Kintsch forecasts that the progress in this field will remain restricted unless a sufficient cognitive theory of writing is developed [Kint87}. This deficit is not surprising because cognitive processes of writing as opposed to reading and comprehension was a largely unexplored field in cognitive science. Although this situation is beginning to change, existing models of writing only emphasize a small section of this complex problem solving activity. We still do not know what is going on in authors' minds when they progress from "chaos to order" as Brown has characterized this process [Brow86].

The widely cited model of Hayes and Flower [Haye80] emphasizes the problem-solving aspect of writing. Based on the analysis of thinking aloud protocols, it identifies three main subprocesses (planning, translating, and reviewing) and their organization in the overall composing activity. Results from experimental research [Kell87] confirm this distinction and indicate that these processes are not subsequent stages but that they show up during the whole course of writing - though at different times with different frequency.

A model which reflects fundamental differences between novice and expert writers has been proposed by Scardamalia and Bereiter [Scar87]. It emphasizes the role of knowledge in the writing process and distinguishes between a knowledge telling and a knowledge transformation strategy. Knowledge transformation is conceived as an interaction between two problem spaces: the content space and the rhetorical space. While the content space is meant to be the space of generating and structuring the author's knowledge about the domain of the intended document, planning and organization of the document structure takes place in the rhetorical space. This is also the place where e.g. decisions on including, excluding, sequencing and reformulating information are made.

Another theory which will bear on the development of authoring systems stems from van Dijk and Kintsch [Dijk83, Kint88]. For our purposes, we will especially make use of their idea of different levels of text organization (micro- and macrostructure) and the corresponding operations (macrooperators) for mapping transitions between them. Whereas this is basically a semantic organization, we will also adopt a more syntactic differentiation proposed by Collins and Gentner [Coll80] between a global text level, a paragraph level (global sequencing) and the sentence or word level (local sequencing). At the global level we will refer to the concept of superstructures [Dijk83]. An example for such a structure is the organization of a scientific report consisting of hierarchically ordered elements, like introduction, method, results etc.

The case of argumentation

A special case of writing is the creation of argumentative texts. Although it seems pretty obvious to relate models of writing to research on argumentation, this is still lacking. On the other hand, there is a long tradition and a variety of schools of thought on what the basic elements of argumentative structures are [Toul58, Ritt72, Wund80, Kopp85]. Systems which are designed to support argumentation have to adopt a specific argumentation model. The gIBIS-system [Conk87] is based on Rittel's ideas of Issue Based Information Systems (IBIS)[Ritt72]. Especially geared to computer-aided reasoning is ARL, a special argumentation representation language, proposed by Smolensky et al. [Smol88] which is used as the basis for the development of EUCLID - a system meant to support argumentation. Argumentation has been investigated to some extent - though mostly with focus different from ours. One example is the OpEd-system [Alva86] an implementation of a model of argument comprehension.

For the representation of arguments, we adopt a schema proposed by Toulmin [Toul58]. Figure 1 shows all different elements of a complete argument. While 'datum' and 'claim' are obligatory constituents, 'warrant', 'backing' and ' rebuttal' are optional. The relation 'so' links a datum and a claim, constituting the following argument which states: "The farmer who does without fertilizer and herbicides in the field and without hormones and tranquilizers in the pigsty has to work much harder than a chemistry farmer" --so--> "it is not worthwhile to produce natural food". Since we are dealing with common sense argumentation instead of formal logic reasoning, an argument is more readily accepted if one can provide a 'warrant' which legitimates the 'so' relation via the 'since' relationship. The warrant provides a general rule which justifies the 'so' conclusion. In a further step, this 'warrant' can be backed by a 'backing' giving evidence for the validity of this rule. In order to handle exceptions from the rule one can use the element 'rebuttal' which questions the claim. Although the original version of the Toulmin schema does not account for the concept of a backing for the 'unless' relationship, our analysis shows that this should be included in a complete schema.

Whereas the Toulmin schema provides an analysis at the micro level, we also need a representation of argumentation at the macro level. Kopperschmidt proposes a hierarchical organization of argumentation resulting in different levels of abstractions [Kopp85]. We will adopt this idea but use our own abstraction hierarchy which shares some featureswith the IBIS-approach [Kunz70, Conk87b] and with subsequent refinements by PHIBIS [McCa83]. This approach differentiates between three types of elements: issues, positions, and arguments. An issue describes the initial question to be answered by the argumentation. The main issues can be divided into subissues. For each subissue one can state at least two positions which can be supported or attacked by a number of arguments.

Cognitive compatibility, externalization, and activity spaces

As indicated before, the idea of cognitive compatibility [Stre87] is our prime principle and guides our system design. This implies that the environment offered to the author corresponds to properties inherent to different cognitive activities and structures of writing. Specifically, we assume that providing different representations which allow easy mapping of internal structures to external task structures and vice versa is a fundamental prerequisite for task-oriented system design [Stre88a]. Second, we adopt the principle of "externalization". Here, we argue that different skills and additional knowledge can be brought to bear on external representations than on internal ones. External representations are open to modification and reinterpretation in more tranparent ways than internal representations. This results in the guideline to provide means which enable the author to externalize as many internal or mental states and intermediary products as possible. Implementing this principle provides the author with different means for structured "thought dumping". Thus, externalization reduces mental load, especially memory load, which results in overcoming the limits of internal representations.

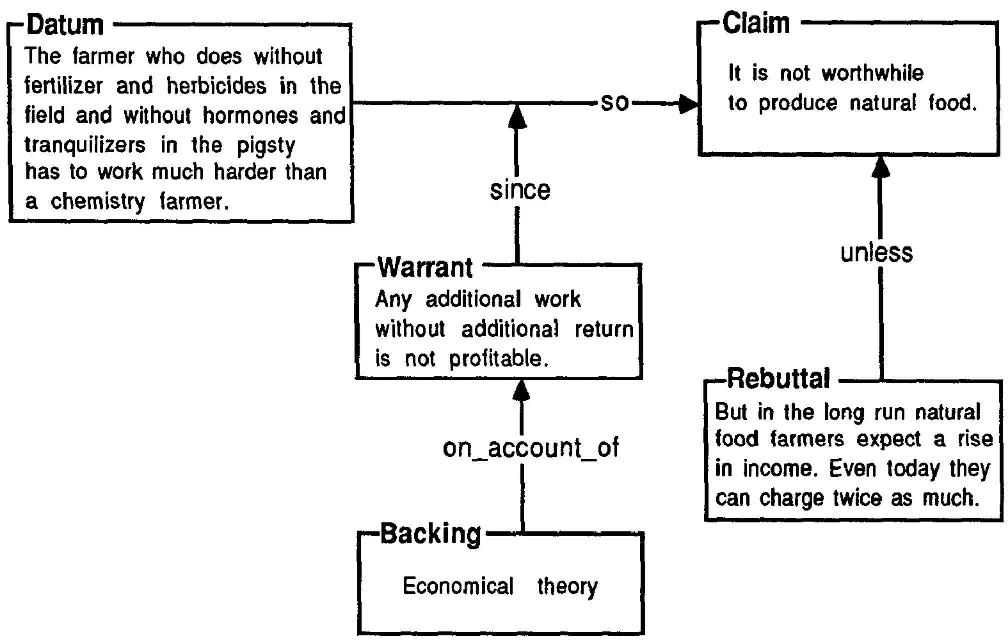


Figure 1. Example of a Toulmin Argumentation Schema

Observation of authors shows for the situation of outlining with paper and pencil that they make extensive use of scribbles and drawings with idiosyncratic notations in order to express part of their internal structures. A standard text editor does not provide a space to do this. There, the author has to write words and sentences - line by line. Usually, something like a scratchpad is not provided. And if one has something like it, it only offers standard graphical tools. But our argumentation is not aiming at providing sophisticated graphic capabilities. It is rather the idea to provide room for specialized notational schemata consisting of elements and relations (leading to node-link hyperstructures) and providing operations on these structures which correspond to generic mental operations of writing and arguing.

This goal is achieved by providing a variety of "activity spaces" realized by dedicated windows which differ in their structural setup and their inherent functionality. The number of activities supported and the extent of this functionality is identified on the basis of cognitive models of writing and task-specific features. Examples are modes for generating and structuring ideas for the content domain (content space), for the type/structure of the target document (rhetorical space), and the style/procedure of argumentation (argumentation space). Our idea of activity spaces originates from Newell's [Newe80] extension of the problem space to be the fundamental organization unit of all cognitive activities and the notion that more than one problem space can be generated during problem solving [Kant84]. Accordingly, we decompose the overall writing activity into a number of specific activities and assign a special space to each of them. Elements of activity spaces are not problem states. Instead, they function as objects of the problem solving activity and are presented to the problem solver who can manipulate them directly. This conceptualization results in a design which has some resemblance to the "Rooms"- metaphor of Card & Henderson an interface which supports fast task switching [Card87]. Furthermore, the idea of activity spaces is similar to the concept of having different windows for different cognitive modes in the WE-system [Smit88]. By distinguishing between network mode, tree mode, edit mode, and text mode, WE focusses mainly on the stages of preparing traditional (linear) documents. Our approach stresses additional cognitive features of the authoring activity, e.g. planning, argumentation, and rhetorical transformations.

Design specifications for activity spaces

Decisions about the number and the functionality of our activity spaces are based on a rationale by integrating results from different models of writing and text production[2]. Details and specifications of these spaces follow therefrom.

First, we adopt the general idea of Scardamelia & Bereiter and distinguish between a "content space" and a "rhetorical space" and corresponding activities [Scar87]. We expand their notion by viewing these spaces as two instantiations of our more general idea of activity specific problem spaces.

Second, we employ Hayes & Flower's analysis of identifying at least three main subprocesses of the writing activity: planning, translating, and reviewing, especially their differentiation of the planning process [Haye80]. Since planning is central to each phase of writing and its results coordinate and guide all other subprocesses, authors need an opportunity for externalization, monitoring and revision of their plans and goals whenever necessary. Therefore, we propose a third space: the planning space.

Third, we have to take into account that our specific activity and document, i.e. argumentation, requires a separate "argumentation space" as a platform for constructing networks of argumentations. This activity is different from generating and structuring elements of the content domain of the intended document and is different from organizing the structure of the document in terms of rhetorical decisions. In summary, our system consists of four spaces which are shown in figure 2.

Although each activity space is defined by its specific characteristics, there are still some common features to all of them. These are derived from our analysis of invariant features of the activities to be supported. First, we provide some generic operations: creating, deleting, copying, naming, renaming nodes and links. Second, it is possible to activate and "open" a node - one way to implement this is the standard use of mouse clicks - which results in the creation of a window to be used for writing and editing the content (e.g. text) of the node. On the other hand, links are also objects which can be activated and edited. The "content" of a link dependends on its type. Example: activating and opening the 'so' link in the argumentation space results in displaying the warrant and backing structure of this link.

---

[2] We are also developing an integrated cognitive model of writing, but there is no space in this paper to give more details of it.

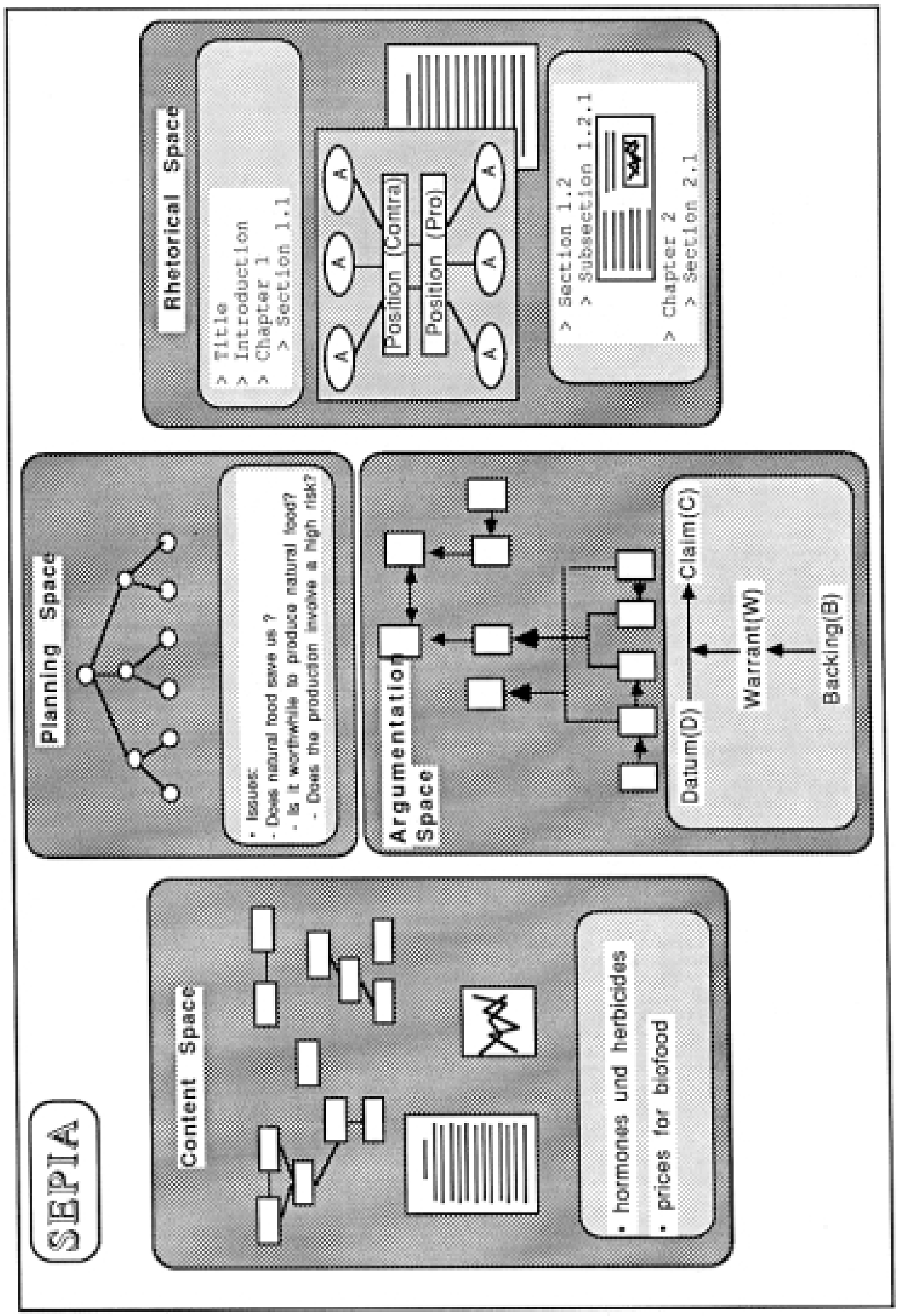


Figure 2. Activity Spaces

Furthermore, we introduce in all spaces the concept of "level". For example, subgraphs of the network in the argumentation space are embedded in a hierarchy thus reflecting different degrees of abstraction which are typical for argumentative texts. On the other hand, there are other relationships within each level so that the total structure is non-linear. The level-concept is motivated for text-like representations by the distinction between micro- and macrostructure proposed by van Dijk & Kintsch [Dijk83] and encompasses also "superstructures" which indicate organizations for larger documents adhering to specific sequencing of chapters and paragraphs.

The following subsections now provide a description of the four spaces: planning, content, argumentation, and rhetorical space. Due to the limited size of this paper, we will only sketch the planning and the content space and rather concentrate on the argumentation and the rhetorical space which are at the center of our current research.

Planning Space

This space serves to support the author in setting up an agenda, in coordinating the whole authoring activity which again requires that the author keeps track of what he is doing (personal monitoring). The function of this space can best be characterized as supporting but also stimulating the authors meta-planning activity. One keystone for the development of a global structure of an argumentative document is the specification of the main issue ("Does natural food save us ?") in a hierarchy of subissues ("Is our food noxious?", " Is it worthwhile to produce natural food?") (see figure 2). These subissues are then taken as topics for working on an argumentation network in the argumentation space. The agenda of interesting questions is not only a planning device for idea generation but at the same time a structured list of topics which can be transformed in the structure of chapters to be used later in the rhetorical space. The planning space functions as a switchboard between the other three spaces. Thus, it contains the overall goal structure and plans for writing. Although the author might stay with his original intentions for quite a while, he is free to change and modify his initial decisions in the sense of what has been called "opportunistic planning" [Haye79].

Content Space

Having identified the domain of the intended document, the author turns - at some point - to the content space. In this space he acquires and collects information about the selected domain which is indicated by the issues identified in the planning space. This can be achieved in two ways. First, the author can start to generate ideas about the domain at a concept level, relate them to each other, e.g. as part-whole relationships, and structure them similar to a semantic network (see figure 2). The purpose here is to obtain a representation of the objects and their mutual relationship involved in the domain of the document.

Second, the author can access additional information which might be documents produced by him before or - sometimes more relevant - which stem from external sources, as e.g. fact and bibliographic data bases, on-line encyclopedias and multi-media knowledge bases. The latter information consists of more or less complete documents where the author can copy parts and use them in the same or in a modified form in his new document. It might also be the case that he receives some stimulation for a new chain of arguments he has not thought of before. Again, this space exhibits different levels ranging from short notes on an idea and sketches of semantic networks to complete multimedia documents which can be viewed and used for inclusion in the intended document. One can think of it as a quarry elements of which are used for further processing by the author.

Argumentation Space

The argumentation space serves as the medium for generating, ordering, and relating arguments for specific issues working at one issue at a time. The representation of the argumentation space in figure 2 provides an impression of the overall structure of this space.

The argumentative activities result in a network having different levels of abstraction. Figure 3 shows an example of an argumentative network with three levels.

The following list provides an overview of possible node and link types as well as of the operations that can be used for their creation and modification in the argumentation network.

Nodes: statements (attributes: name, position, claim, datum, level)

Links:

- so (attributes: warrant, backing)
- contradicts
- contributes_to

Operations:

- generate
- support
- object_to
- justify
- negate
- generalize
- specialize

Nodes and links

The nodes of the network represent statements generated by the author during the development of his argumentation. Each node is an object which is characterized by four attributes, we call 'position', 'claim', 'datum', and 'level'. The values of the first three attributes are boolean values relative to the author's subjective representation. Compared to the model of argumentation proposed by Toulmin, datums and claims are both statements but differ in their attributes. In figure 3, 'setbacks' represents a statement at the medium level of the net which is the datum for 'losses' at the same level. Let the content of the first node be 'organic farmers often are victims of disastrous setbacks' and the content of the second 'organic farmers often have to face grave financial losses' then both nodes together with the 'so'-link constitute a simple argument according to Toulmin's model. As can be seen at the lowest level of the net, authors can go beyond such simple arguments by developing argumentative chains the intermediate elements of which simultaniously function as datum and claim, i.e. they can take different roles. At the top level of the net there are two nodes which represent positions. We consider positions as general claims like 'it is worthwhile to produce natural food'.

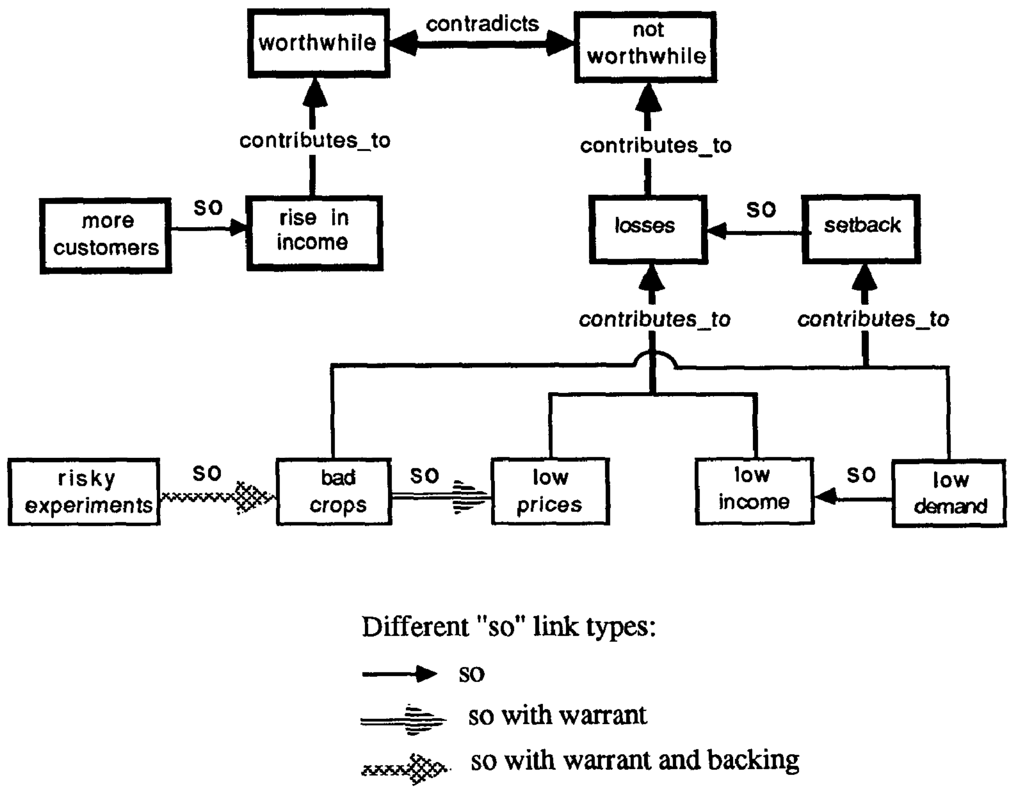


Figure 3. A network resulting from argumentation

Beside the relation 'so', the relation'contradicts' can be used to connect two statements at each level of the network. But contrary to 'so', this link always connects statements of the same type, i.e. a datum with a datum, a claim with a claim, and a position with a position. It therefore describes the relationship of mutual opposition. For example, in figure 3 the position 'worthwhile' directly contradicts the position 'not worthwhile'.

Relations between different levels of the net are represented by the relation 'contributes_to'. For example, in figure 3 the claim 'losses' contributes to the position 'not worthwhile', which means that the statements 'organic farmers often have to face grave financial losses' contributes to the position 'it is not worthwhile to produce natural food'.

Operations within levels

So far, we have described elements of the argumentative structure but we have not yet specified how this structure is built up. Therefore we have to describe the different operations an author can apply in the course of arguing. To start with, the most basic operation is the generation of a statement. Whenever the author chooses this option a new, unrelated node appears in the argumentation space which has to be named and then can be further processed. For assigning a statement to a new node the author has to use the operation 'edit'. When 'edit' is applied to the node, a text window appears on the screen and the author can write down the intended statement. By default, each node is a claim and functions as a position as long as no other operations have been applied to it.

Statements in the space can be supported by other statements. The operation 'support' results in a new link, i.e. the relation 'so' is established, leading from the supporting to the supported statement. Since the supporting node now serves as a datum its corresponding attribute is changed accordingly. By applying the operation to nodes which are already related to other nodes at the same level an author is able to construct argumentative chains. In figure 3 such a chain is shown at the lowest level.

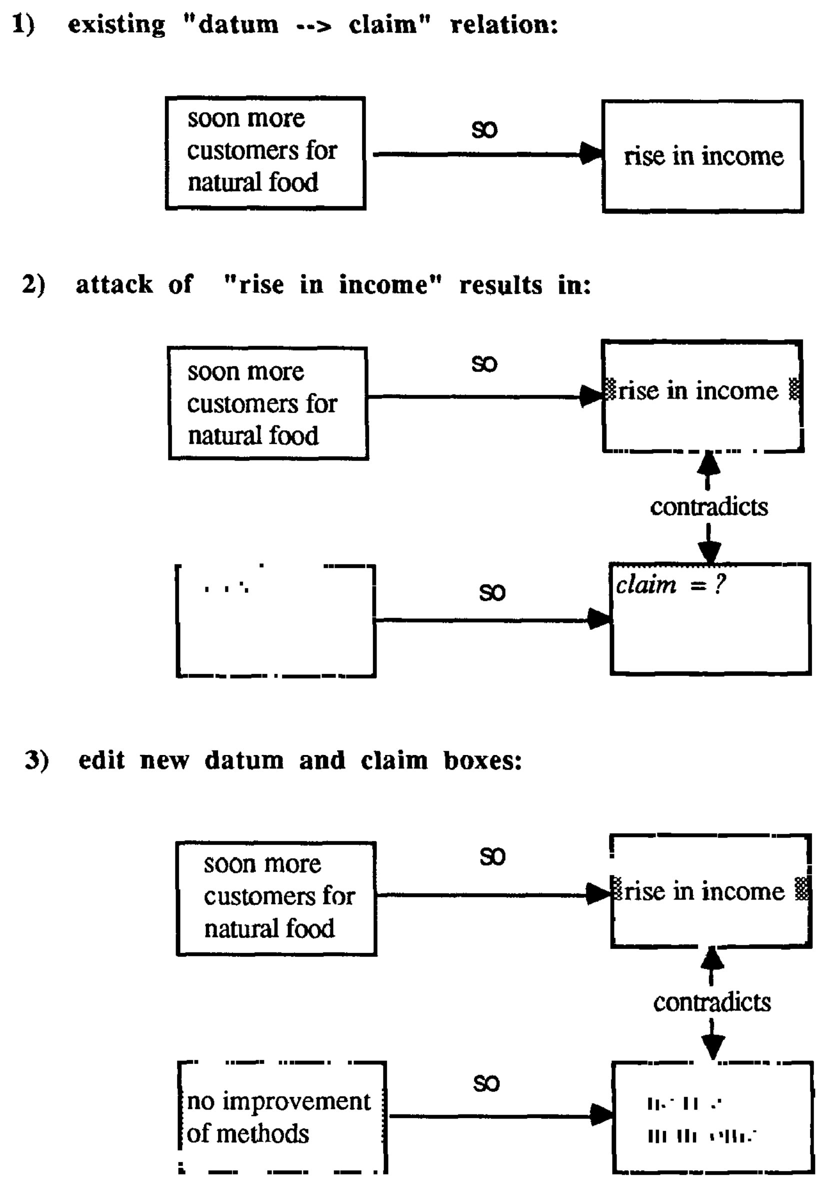


Figure 4. Steps of the operation 'object_to'

If an author wants to indicate that a statement is the direct contradiction of another statement, he can achieve this by using the operation 'negate'. The operation constitutes the relation 'contradict' which connects the two opposite statements. It can be applied at all levels of the network, even at the top.

A more complicated situation arises when the author wants to attack a claim. He can do so by using the operation 'object_to'. Figure 4 shows how the operation works. Suppose the author wants to attack the claim of the argument displayed in part (1) of the figure. When he applies 'object_to' to the claim the system creates two nodes related by 'so' and then connects the claim of this new argument via 'contradict' to the claim under attack. The result is shown in part (2). The author can now formulate his objection by editing the new nodes. This leads to the final structure displayed in part (3) of the figure.

It is obvious that 'object_to' generates a more complicated argument than originally intended by the author. The reason for this is to facilitate further arguing by making the implications of an attack explicit: if a claim can be attacked by a datum the same datum can be interpreted as support for the negation of the claim. Demonstrating this relationship to the author by automatically generating the structure in part (2) should lead to a more complete view of the consequences which are entailed by his argumentation.

The relation 'so' can further be specified by the operation 'justify'. This operation provides the author with the opportunity to formulate a warrant and a backing to cover the relation between datum and claim. When 'justify' is used a new window appears on the screen in which a schema is displayed. The schema shows the model of argumentation proposed by Toulmin and is presented as a template consisting of four slots and three links (see figure 2).

The slots named 'datum' and 'claim' of the schema contain the statements of the two nodes whose connection shall be justified. The author can now fill in the other two slots by generating appropriate statements which function as warrant and backing. The modification of the link leads to a change of its attributes which in turn is indicated in the network by a new symbol for the 'so'-relationship. The 'so' between 'bad crops' and 'low prices' in figure 3 represents a link which is justified by a warrant alone, whereas the connection between 'risky experiments' and 'bad crops' stands for a 'so' which is further covered by a backing.

Operations between levels

All operations described so far serve as means to incrementally develop and expand arguments at a given level of abstraction and we believe that a good deal of the structural characteristics of argumentative documents can be modelled by them. However, in many cases only one level of abstraction is not sufficient for representing the structures which arise in the course of a more complex argumentation. For example, authors often start with a global argument, as indicated at the medium level in figure 3, and then gradually refine it by creating other arguments which are specific instantiations of it. Another way to proceed is to write down several arguments and then draw a specific conclusion by summarizing their claims in a more general position. To support authors in those activities of organizing, the argumenation space provides them with two operations which establish different levels of abstraction in an argumentation. Using the operation 'generalize' an author can create a statement and relate it to one or more statements via a link labeled 'contributes_to'. If the specific statements are situated on level i the more general statement gets located at level i+1. The level of a statement is reflected in its 'level' attribute as well as in its position in the space: general statements appear on a higher level than more specific ones. The counterpart of the operation 'generalize' is 'specify'. It enables the author to create more specific statements and link them to the appropriate general statements.

Both operations can be used in a variety of ways thus creating argumentative structures of different hierarchical complexity. For example, if the author wants to summarize several specific arguments in a more general one, he can accomplish this by first generalizing their datums and claims and then using the operation 'support' to link these generalizations. The left part of figure 3 shows a structure which has been developed this way. On the other hand, the author can first write down a more global argument and then start specifying it by more concrete arguments. Thus the combination of both 'generalize' and 'specify' guarantees a high flexibility in developing, organizing, and structuring a complex argumentation.

Another way of using the two operations lies in the generation or specification of general positions. As described above, a position is a general claim. If the author develops his argumentation in a top down manner he may first generate such a claim and then use the operation 'specify' to create arguments which favour it. On the other hand, if he prefers to argue bottom up he may reach the same goal by starting at a low level and using the operation 'generalize' to join the claims of his specific arguments in a general position. Since both ways are often used the combination of 'specify' and 'generalize' provides authors with a high flexibility with regard to their strategy of argumentation.

The data types (nodes, links) as well as the operations described in this paragraph form a useful basis for supporting authors in their attempt to construct sophisticated argumentations. The operations in the space obey the principle of cognitive compatibility and the proposed network gives a clear view of the argumentative structure preserving the different levels typically found in more complex argumentations. Therefore, we think that this space considerably facilitates the writing of argumentative documents.

Rhetorical Space

In essence, argumentation is always an interaction or at least a dialog. Writing an argumentative document has to account for the absence of the dialog partner. This situation implies that the objections of an opponent have to be anticipated. Therefore, writing an argumentative text, reflections about the reader are more relevant than in other types of text. The author's intention is not only to communicate his ideas, but also to convince others of the validity of those ideas. To increase the persuasiveness of an argumentation, is not only a matter of content and argument structures but above all a matter of discourse and rhetorical structuring.

By considering rhetorical aspects, the author imposes a document structure on his ideas and arguments which he has explicated and elaborated in the content and argumentation space. For this task we propose another separate activity space, called rhetorical space. The problems to be solved in this space require:

- decisions about the global outline of the document
- decisions about the rhetorical reorganization of positions and arguments for each subissue
- decisions about writing coherent sentences

To support authors in making these rhetorical decisions, we distinguis between three modes: an outline mode, an argumentation strategy mode, and a text edit mode (compare figure 2).

Outline mode

In the planning space, the author has already decided on the issue hierarchy. Now he has to decide when and where to deal with these issues. In order to use the tree structure of subissues for the generation of an outline format of the document, the author selects an issue, a subtree or the whole tree and copies it in the outline mode. As a result of this operation, he gets an outline structure which is completely or partially identical with the hierarchy of issues. This means that either a substructure of the issue tree is mapped in a one-to-one correspondence on the hierarchy of chapters or that some selected issues are combined in one chapter or distributed over several chapters.

In most cases, argumentation is embedded in a context. Thus, the document has to include sections which have no direct relation to any of the specified arguments. Therefore, the author can expand the outline structure by creating a new headline with a link to a text node previously generated in the content space. Establishing a context for argumentation can be supported by providing a conventionalized text schema. Van Dijk and Kintsch call such a schema the superstructure of a text [Dijk83]. An example of a superstructure which is relevant for the domain of argumentation is the usual outline of a scientific report. If there exists such a convention for a certain type of document, the system will offer the corresponding superstructure as a guideline for the development of the outline structure.

Argumentation strategy mode

Beyond establishing text coherence at a global level, the author has to arrange positions, counterpositions, and arguments into a convincing form. We call this important rhetorical task the development of an argumentation strategy.

For the development of this strategy the system provides a special environment in a subwindow. The rhetorical reordering starts with a selection of an argumentative subnet to be copied from the argumentation space. The selection of subnets is guided by the selection of positions, which are relevant for a particular subissue. The selected positions are not copied together with all arguments but only with those from the next lower level of the abstraction hierarchy. Now the author has the freedom to reorder the relation of arguments and positions. On the other hand, there are traditional rhetorical schemata, as e.g. the top-town form (pyramid form) with the position slot on top, or the bottom-up form (inverse pyramid form) with the slots for the arguments first and then the supported position (see figure 2). One possibility of support is that a selected argumentation structure is copied into this schema and the slots are filled with the name of a position and its corresponding arguments.

By applying some basic rhetorical operations the author can change the sequence of the statements or delete some elements of the argumentation structure.

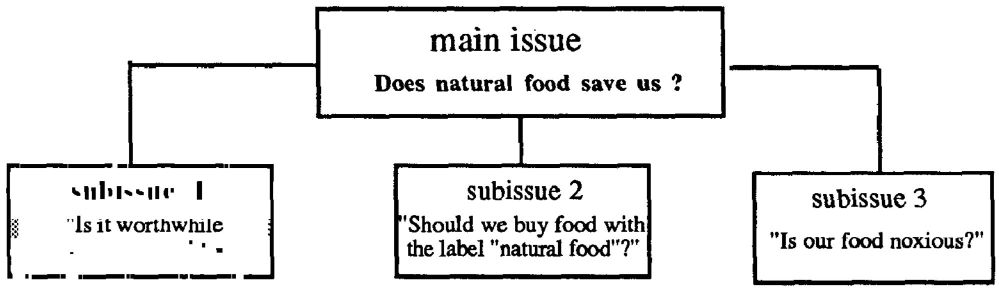
delete: Not every statement in a selected argumentation structure is worth to be incorporated in the final document. In order to leave some obvious positions and well known arguments implicit, the author can delete specified elements.

reorder: The order of arguments is very important for the development of an argumentation strategy, because the same arguments may be of different importance for different groups of readers. If an author wants to consider these differences, he can change the order of arguments with the reorder command, thereby adjusting his argumentation to the reader´s profile.

expand: Since the argumentative subnet selected by the author includes only a position and the next lower level of the argumentation hierarchy, the author needs support if he wants to expand this structure. By activating an argument and applying the expand operation, he gets those arguments which are connected to the activated argument on the next lower level This type of expansion is a node expansion. Another form is to expand a link. Activation of a 'so'-link returns warrant and backing justifying the corresponding 'so'-relation as an additional hypertext node.

The results of these operations are demonstrated in figure 5. It represents a section of an article dealing with the main issue "Does natural food save us?" and corresponds to the previously described argumentation net in figure 3. Part 1 of figure 5 shows a section of the issue structure of the article, part 2 describes the rhetorical structure of one of the subissues.

1) Part of an issue structure



2) A rhetorical structure for subissue 1

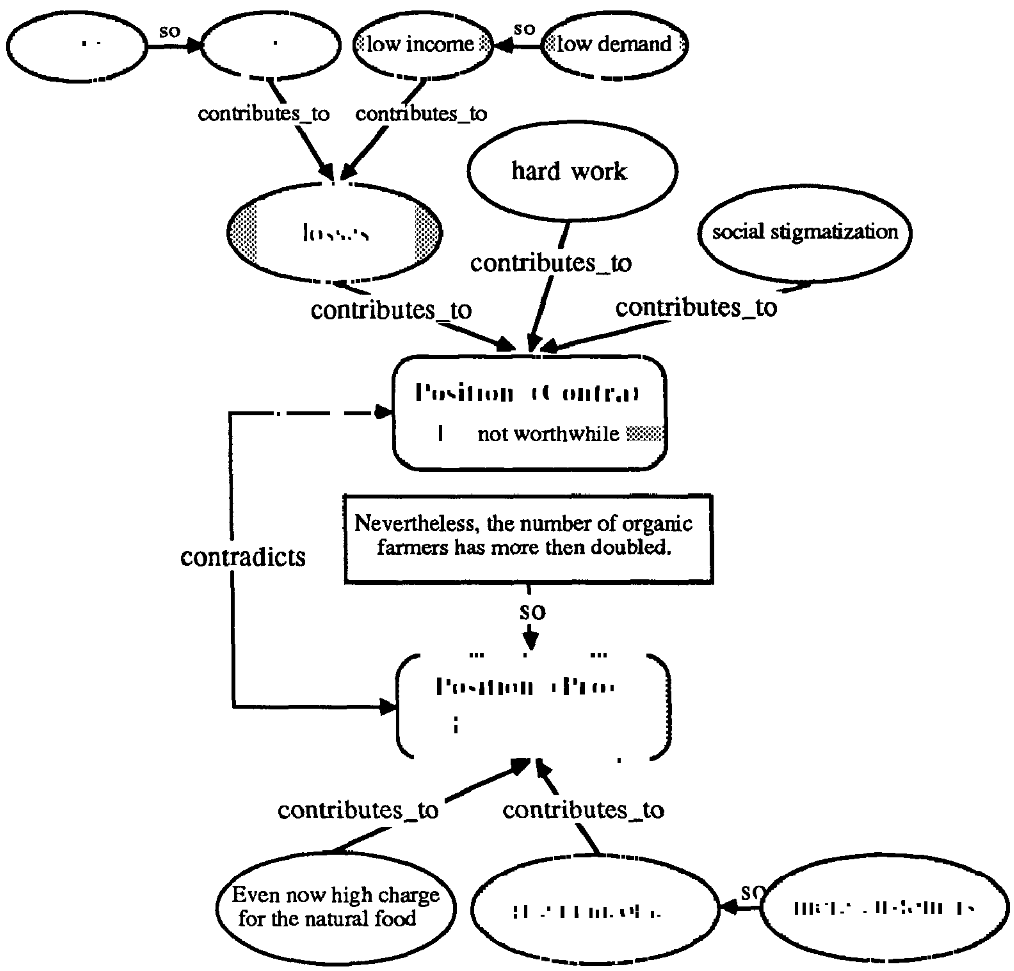


Figure 5. Example of a rhetorical structure

As the rhetorical space provides means to reorganize and enrich selected parts of the argumentative network and to place them in a rhetorically rearranged hierarchy of subissues, different types of documents can be produced on this basis. We only mention two examples: (1) a "guided tour" through the argumentation network. Starting with a selected issue, the recipient is guided through the original line of the author's argumentation leading to a spelled out position. (2) In general, the result of the rhetorical transformations will be a coherent full text document having either a linear or a non-linear structure with respect to the "surface" information about facts and statements. In either case, the document is a hyperdocument providing not only the factual information for the reader but also access to the underlying argumentation structure. Therefore, the reader gets - even while reading a "traditional" document - the exciting possibility to go back to the sources, i.e. to early and intermediate products of the authoring process ("travelling back to the quarry"). This way, the reader can duplicate the author's argumentation at a more detailed level. In a next step of our research, we will specify the functionality necessary for the production of these types of hyperdocuments.

Text edit mode

In every mode, the author has a text editor at his disposal which allows him to create the content of a new node, to edit the content of existing nodes, and to write transitions so that the final document meets the criteria of coherence, connectivity and fluency.

TRAVELLING ACTIVITY SPACES

Empirical research on the writing process of mature authors has shown that production and transformation of knowledge during writing is not done in a linear sequence of stages from idea generation to text generation. There is a constant interaction between levels of text representation as well as between different subtasks. For instance, problems in formulating an argument clearly and convincingly might result in the subgoal "generate an example illustrating this argument". Let us translate this situation into our world of activity spaces. Although the rhetorical space is always the final destination, an author's train of thought is not heading there in a straightforward way. Instead, we expect heavy traffic between the spaces. This resembles the interaction of cognitive processes when locating the solution of different subproblems in different problem spaces which can be mapped to the proposed activity spaces.

> Information flows from the *planning space* to the other three spaces: Issues specified in the planning space set topics for the content space, direct the structuring in the argumentation space, and are transformed into an outline in the rhetorical space. On the other hand, information flows from these three spaces back to the planning space: Operations in each of them might result in new insights leading to the formulation of new goals or to the specification of new subissues in the planning space ("opportunistic planning").
>
> Structured knowledge, elicitated or generated in the *content space* may be translated into argumentative statements in the argumentation space or into a nonargumentative text passage in the rhetorical space.
>
> There is a close relationship between the *argumentation space* and the rhetorical space because the argumentation space functions as a quarry for the rhetorical space. The argumentation graph or parts of it must be reorganized according to an argumentation strategy in the rhetorical space. In addition, the need for explication of an argument often results in operations in the content space discovering new insights and thus reducing the original fuzzyness.
>
> The detection of deficits in the *rhetorical space* leads to the formulation or specification of new arguments in the argumentation space. It may also require certain operations in the content space, such as searching for missing knowledge or elaborating facts generated so far.

EMBEDDING ACTIVITY SPACES IN SEPIA

The activity spaces are the environment where authors generate and structure their ideas. But this is not the whole story. As we have mentioned in the introduction, the idea of activity spaces is central but it is only one component of a more comprehensive system called SEPIA: Structured Elicitation and Processing of Ideas for Authoring. Although this paper only addresses the concept and design requirements for activity spaces, we want to place these ideas into a wider perspective. Therefore, we provide in this section an overview of the complete system we are currently developing. Beyond the basic and central functionality of the Generation/Structuring Component, we will realize an active part for the SEPIA-system including the following components (see figure 6).

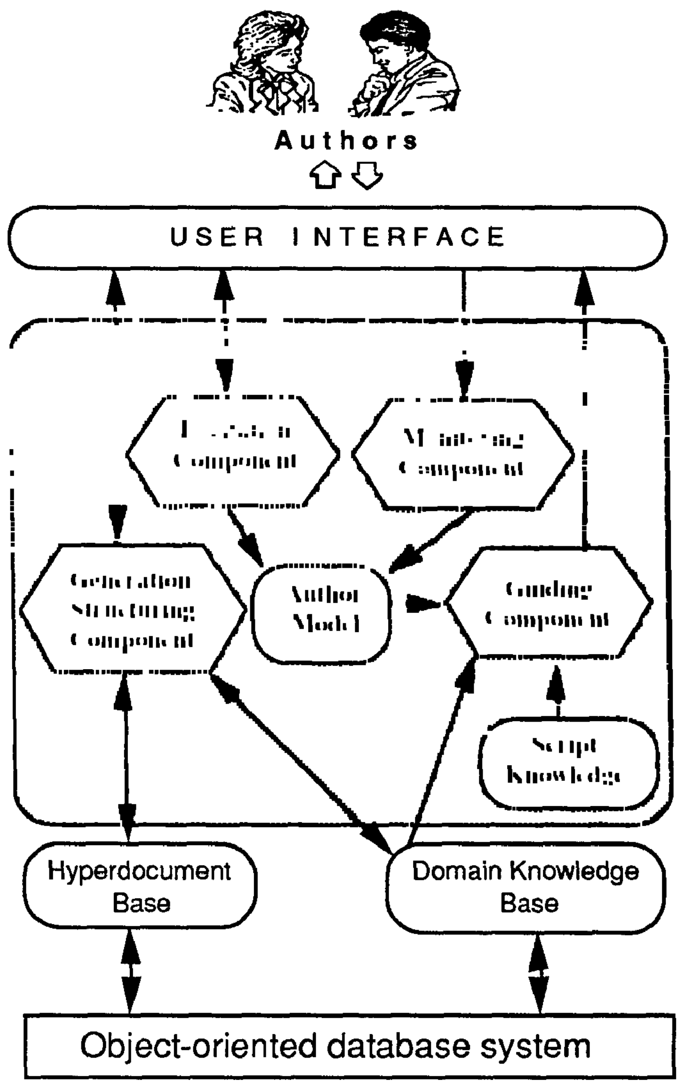


Figure 6. Architecture of the knowledge-based authoring tool SEPIA (Structured Elicitation and Processing of Ideas for Authoring)

First, in order to provide feedback and to react in an author-oriented way the system has to acquire knowledge about the author's original goals and plans, intended topic domain, document type, writing strategy, target group, etc. These data are acquired via a Knowledge Elicitation Component employing an interactive dialog technique. They form the basis of the system's model about the author (the initial author model).

Second, in order to be active at an appropriate point of time, the system has to monitor the author's writing behavior. The Monitoring Component provides protocol data about the author's activities in different windows (= activity spaces). These data are used to update a dynamic part of the author model which obviously has to be distinguished from a rather static part corresponding to the author's initial profile. In order to use this information in a coherent way for guiding we will provide a mechanism which integrates the data coming from different information sources. A future goal for the monitoring component is to make use of results from research on text analysis in order to employ a semantic analysis of the author's input.

Third, a Guiding Component processes the integrated information about the author's activity (i.e. the dynamic author model), compares it to information about the earlier acquired author's profile (on goals and plans, etc.) and to information stored in a script knowledge base. The script knowledge base contains knowledge about document types (e.g. hypertext structures) and argumentation structures. In a more advanced version, the system will also use additional knowledge about the content domain. Thus, the system has "objective" knowledge of rules of discourse (argumentation) and rules and facts of the domain. On the other hand, it has "subjective" knowledge of the author: i.e. of the author's mental model of the domain and the author's mental model of discourse esp. argumentation. Based on the result of the comparison of the objective knowledge to the subjective one the system provides feedback and active support/advice to the author following a specific guiding strategy. Of course, these additional features need much more detailed considerations which will be reported elsewhere.

CONCLUSIONS AND RECOMMENDATIONS

In this section, we summarize our ideas within a more global framework and make some recommendations for the design of future hypertext systems. The availability of innovative technology for the production and reception of electronic documents raises the question of how to improve the quality of these documents. There are at least two aspects to the notion of quality. The first refers to additional multi-media features, as e.g. high-resolution graphics and images, animation and simulation, video including sound, etc.

The second is the extent to which the document contains additional structural information which can be used for further processing. Although multi-media features are of course desirable for hyperdocuments, we want to stress the second aspect.

The current situation is characterized by predominantly linear documents - on paper as well as in electronic format. An additional level of structure can be found in currently produced hyperdocuments. Unfortuately, in most cases it is restricted simple "points_to" links. A general hypertext model should go beyond this and rather employ the full capability of multiple types of nodes and links, including composites as an augmentation of the basic node and link model [Hala88]. The analysis and specification proposal in this paper has identified how one can enrich hyperdocuments with special links and nodes. We have demonstrated this for argumentative and rhetorical structures. The enrichment of hyperdocuments is reached by preserving structural information created and used by the author along his composing activity. The information concerns the final structure of the document as well as that of transient intermediary products. It has to be noted that this additional structure can be accessed by the author as well as the reader. It thus facilitates the comprehension of the intended message. Another advantage of this approach is that these links are also machine processible.

Additional information of this kind is only available if the author is provided with tools which allow him to externalize his internal structures by making them part of the hyperdocument. Since the use of these tools should not result in an additional burden we have to design them in a user-oriented and task-driven fashion in accordance with the principle of cognitive compatibility. In summary, this leads to the conclusion that developers of hypertext systems should focus on the role of authoring more intensively than they do today.

Such a demand is intrinsically linked to the issue of human factors aspects for interface design of hypertext systems. Our approach led to the concept of activity spaces and the heavy utilization of externalizing subjective knowledge structures based on the author's mental models. Activity spaces implemented as dedicated windows with activity specific functionalities can be viewed as a natural metaphor for supporting cognitive processes in any interactive problem solving activity. Thus, this concept is applicable to a wide range of interactive systems and not restricted to hypertext systems.

A further and logical step in the development of authoring tools in hypertext environments is to extend the support by knowledge-based components as given in the description of the SEPIA-system. It has to be pointed out that this kind of support is possible because monitoring of the author's externalized behavior provides the system with information that can be used to guide his subsequent activities.

A final remark

If our argumentation for designing an authoring system this way did not convince you to the extent we hoped it would, this might be due to the circumstance that - unfortunately- the system we propose is still under development and therefore has not been at our disposal for writing this paper.

Acknowledgements

Thanks are due especially to our colleagues Jörg Haake, Werner Rehfeld, Helge Schütt, and Wolfgang Schuler working in the WIBAS-project. The ideas presented in this paper have greatly profited from our dicussions with them and their stimulating comments.

REFERENCES

[Aksc88] Akscyn, R.M., McCracken, D.L. & Yoder. E.A. KMS: A distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 31(7), 1988, 820-835.

[Alva86] Alvarado, S.J., Dyer, M.G., & Flower, M. Editorial comprehension in OpEd through argument units. In: *Proceedings of the 5th National Conference of Artificial Intelligence - AAAI '86* (Vol. 1). Los Altos, CA: Kaufmann, 1986, pp. 250-256.

[Brow86] Brown, J.S. From cognitive to social ergonomics and beyond. In D. Norman, & S. Draper (Eds.), *User-centered system design: New perspectives on human-computer interaction.* Hillsdale, N.J.: Erlbaum, 1986, pp. 457 - 486.

[Brow87] Brown, P.J. Turning ideas into products: The Guide System. In: *Proceedings of the Hypertext '87 Workshop*, Chapel Hill, NC, 1987,.pp. 33-40.

[Card87] Card, S.K., & Henderson, A. A multiple, virtual-workspace interface to support user task switching. In Carroll, J.M., & Tanner, P.P. (Eds.), *Proceedings of the CHI und GI '87 Conference on Human Factors in Computing Systems*, Toronto. New York: ACM, 1987, pp. 53-59.

[Coll80] Collins, A., & Gentner, D. A framework for a cognitive theory of writing. In L.W. Gregg, & E. Steinberg (Eds.), *Cognitive processes in writing: An interdisciplinary approach* Hillsdale, NJ: Lawrence Erlbaum, 1980, pp. 51-72.

[Conk87a] Conklin, J. Hypertext: An introduction and survey. *IEEE Computer Magazine*, 20(9), 1987, 17-41.

[Conk87b] Conklin, J., & Begeman, M.L. gIBIS : A hypertext tool for team design deliberation. In: *Proceedings of the Hypertext '87 Workshop*, Chapel Hill, NC, 1987, pp. 247-251.

[Deli86] Delisle, N.A., & Schwartz, M.D. Neptune: A hypertext system for CAD applications. In C. Zaniolo (Ed.), *Proceedings of the International Conference on Management of Data - SIGMOD '86*, Washington, D.C. New York, NY: ACM, (1986), pp. 132-143.

[Dijk83] Dijk, T.A. van, & Kintsch, W. *Strategies of discourse comprehension.* New York, NY: Academic Press, 1983.

[Hala87] Halasz, F.G., Moran, T.P., & Trigg, R. Notecards in a nutshell. In J.M. Carroll, & P.P. Tanner (Eds.), *Proceedings of the CHI und GI '87 Conference on Human Factors in Computing Systems,* Toronto. New York: ACM,1987, pp. 45-52.

[Hala88] Halasz, F.G. Reflections on Notecards: Seven issues for the next generation of hypermedia systems. *Communication of the ACM*, 31, 1988, 836-852.

[Haye80] Hayes, J.R., & Flower, L.S. Identifying the organisation of writing processes. In L.W. Gregg, & E.R. Steinberg (Eds.), *Cognitive processes in writing*. Hillsdale, NJ: Lawrence Erlbaum 1980, pp. 3-30.

[Haye79] Hayes-Roth, B., & Hayes-Roth, F. A cognitive model of planning. *Cognitive Science*, 3(4), 1979, 275-310.

[Kant84[Kant, E. & Newell, A. Problem solving techniques for the design of algorithms. *Information Processing & Management*, 20(1-2), 1984, 97-118.

[Kell87] Kellog, R.T. Effects of topic knowledge on the allocation of processing time and cognitive effort to writing processes. *Memory & Cognition*, 15(3), 1987, 256-266.

[Kint87] Kintsch, W. Foreword. In C. Bereiter, & M. Scardamelia, *The psychology of written composition.* Hillsdale, NJ: Lawrence Erlbaum, (1987), pp. 9-12.

[Kint88] Kintsch, W. The role of knowledge in discourse comprehension: A construction integration model. *Psychological Review*, 95(2), 1988, 163-182.

[Kopp85] Kopperschmidt, J. An analysis of argumentation. In T.A. van Dijk (Ed.), *Handbook of discourse analysis: Vol. 2. Dimensions of discourse* London: Academic Press, 1985, pp. 159-168.

[Kunz70] Kunz, W., & Rittel, H. *Issues as elements of information systems* (Working paper 131). Berkeley, CA: University of California, Center for Planning and Development Research, 1970.

[McCa83] McCall, R., Schaab, B., & Schuler, W. An information station for the problem solver: system concepts. In C. Keren & L. Perlmutter (Eds.), *Applications of mini- and microcomputers in information, documentation and libraries.* New York: Elsevier, 1983.

[Marc88] Marchionini, G., & Shneiderman, B. Finding facts versus browsing knowledge in hypertext systems. *IEEE Computer*, 21(1), 1988, 70-81.

[Newe80] Newell, A. Reasoning, problem solving, and decision processes: The problem space as the fundamental category. In: R. Nickerson (Ed.), *Attention and performance VIII.* Hillsdale, N.J.: Lawrence Erlbaum, 1980, pp. 693-71.

[Scar87] Scardamalia, M., & Bereiter, C. Knowledge telling and knowledge transforming in written composition. In S. Rosenberg (Ed.), *Advances in applied psycholinguistics: Vol. 2. Reading, writing, and language learning.* Cambridge: Cambridge University Press, 1987, pp. 142-175.

[Smit87] Smith, J.B., Weiss, S.F., & Ferguson, G.J. A hypertext writing environment and its cognitive basis. In: *Proceedings of the Hypertext '87 Workshop,* Chapel Hill, NC, 1987, pp. 195-214.

[Smit88] Smith, J.B., & Lansman, M. *A cognitive basis for a computer writing environment.* (Technical Report). Chapel Hill, NC: University of North Carolina, Department of Computer Science, 1988.

[Smol88] Smolensky, P., Fox, B., King, R., & Lewis, C. Computer-aided reasoned discourse or, how to argue with a computer. In R. Guindon (Ed.), *Cognitive Science and its application for human-computer interaction.* Norwood, NJ: Ablex, 1988, pp.109-162.

[Stre87] Streitz, N.A. Cognitive compatibility as a central issue in human-computer interaction: Theoretical framework and empirical findings. In G. Salvendy (Ed.), *Cognitive engineering in the design of human-computer interaction and expert systems.* Amsterdam: Elsevier, 1987, pp. 75-82.

[Stre88a] Streitz, N.A. Mental models and metaphors: Implications for the design of adaptive user-system interfaces. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems.* New York: Springer, 1988, pp. 164 - 186.

[Stre88b] Streitz, N.A., & Hannemann, J. *Writing is rewriting: A cognitive framework for computer-aided authoring.* Paper presented at the 4th European Conference on Cognitive Ergonomics (ECCE 4), Cambridge (England), 1988.

[Toul58] Toulmin, S. *The uses of argument.* Cambridge: Cambridge University Press, 1958.

[Trig86] Trigg, R., & Weiser, M. TEXTNET: A network-based approach to text handling. *ACM Transactions on Office Systems*, 4(1), 1986, 1-23.

[Trig87] Trigg, R.H., & Irish, P. M. Hypertext habitats: Experience of writers in NoteCards. In: *Proceedings of the Hypertext '87 Workshop*, Chapel Hill, NC, 1987, pp. 89-107.

[Walk87] Walker, J.H. Document Examiner: Delivery system for hypertext documents. In: *Proceedings of the Hypertext '87 Workshop*, Chapel Hill, NC, 1987, pp. 307-324.

[Will87] Williams, G. HyperCard. *Byte*, 12, 1987, 109-117.

[Wund80] Wunderlich, D. *Arbeitsbuch Semantik.* Frankfurt: Athenaeum, 1980.

[Yank85] Yankelovich, N., Meyrowitz, N., & Dam van, A. Reading and writing the electronic book. *IEEE Computer*, 18(10), 1985, 15-30.