WORKING PAPER

SCDAS - Decision Support System for Group Decision Making: Information Processing Issues

Andrzej Lewandowski

July 1988
WP-88-48

IIASA
International Institute for Applied Systems Analysis

SCDAS - Decision Support System for Group Decision Making: Information Processing Issues

Andrzej Lewandowski

July 1988
WP-88-48

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

Most of research in the field of computerized Group Decision Support Systems is devoted to analysis and support the quantitative phase of decision processes using various methods of multiple-criteria analysis. The experience shows, that the soft side of the decision process needs also certain support. This relates mostly to distribution of textual information which augments the quantitative side of decision process and to providing the linkage between such information and numerical data. This aspect is especially important when the decision support system is implemented in distributed computing environment. In the paper the possible forms of information processed within the SCDAS system are analysed as well as the framework for implementing the software that provides such processing functions is presented.

Alexander B. Kurzhanski
Program Leader
System and Decision Sciences Program

SCDAS - Decision Support System for Group Decision Making: Information Processing Issues

Andrzej Lewandowski

SCDAS – Decision Support System for Group Decision Making: Information Processing Issues

Andrzej Lewandowski
International Institute for Applied Systems Analysis
Laxenburg, Austria *

1 Introduction

The SCDAS system (Selection Committee Decision Analysis and Support) has been designed for supporting such decision problems, where the group of experts (the ***committee***) cooperates to select the best alternative (or to reduce the set of alternatives to some reasonable subset which can be considered for further analysis) among alternatives presented to them by independently acting experts. Detailed assumptions and description of SCDAS procedure is presented in the paper by Lewandowski and Wierzbicki (1987).

Up to now exist several experimental implementations of SCDAS procedure (see Lewandowski, 1988). All these implementations have been prepared mostly to investigate the algorithmic and procedural aspects of SCDAS framework as well as to perform experimental applications of this methodology (for such an experimental application see Dobrowolski at all., 1987). Several questions relating to procedural and algorithmic details of the procedure are still open and more research in this field is necessary. These topics include first of all problems of uncertainty, quality of information, sensitivity analysis, presentation and analysis of results etc.

During experiments with existing prototype implementation of SCDAS as well as experiments with participation of decision makers it became clear, that support of quantitative aspects of decision process must be augmented by tools for supporting qualitative phase of this process. The idea that the ***discussion*** between committee members is one of the most important part of the decision process has been already mentioned in quoted above papers. It was stated by DeSanctis and Gallupe (1987):

> ***"...A group decision occurs as the result of interpersonal communication – the exchange of information among members....The communication activities exhibited in a decision-related meeting include proposal exploitation, opinion exploitation, analysis, expression of preference, argumentation, socializing, information seeking, information giving, proposal development and proposal negotiations...In this sense the goal of GDSS (Group Decision Support System) is to alter the communication process within groups..."***

Huber (1984) also expresses the importance of qualitative support for decision making:

*This paper reports the research performed according to the agreement between IIASA and Digital Equipment Corporation within the European External Research Program, Contract Number AU-008

"...Information sharing is the most typical of the activities in which groups engage... general GDSS can also enable groups to elicit, share, modify and use professional judgements and opinions in at least as many ways as they do hard data..."

Without the consensus related to procedural principles and other important aspects of decision process it is not possible to provide any quantitative support. This consensus can be however reached only after *discussion* and *exchange of information* between committee members.

Therefore, the group decision support system should be treated as *information processing* and *information management* system. According to DeSanctis and Gallupe (1987), within the information exchange view of group decision making systems there are three levels of supporting the group decision process:

- *Level 1* provides technical features aimed at removing common communication barriers, such as large screens for displaying of ideas, voting solicitation and compilation, anonymous input of ideas and preferences, electronic messages between members. Summarizing, *Level 1* systems improve the decision process by facilitating information exchange among members.

- *Level 2* provides decision modelling and group decision techniques aimed at reducing uncertainty and "noise" that occur in the group's decision process. Such tools as risk analysis, multiattribute utility function can be considered as tools for supporting group decision process and offered to group members on this level. Therefore, this level represents an enhanced GDSS as opposed to *Level 1* which provides communication media only.

- *Level 3* can include expert advice in selecting and arranging of rules to be applied during a meeting. This includes active information filtering, adaptation of decision procedure to a given decision problem, etc. Therefore, GDSSs belonging to this group can be named *active decision support systems.*

Definitely, the SCDAS system belongs to the *Level 2*, since it provides well defined procedural framework for decision making and well defined and theoretically backgrounded methodology for quantitative support of decision processes. These aspects of SCDAS approach have been addressed in previous publications (Lewandowski and Wierzbicki, 1987). Similarly, all existing implementations devote only decision-theoretic and quantitative aspect of decision support methodology. The main purpose of these implementations was to validate the quantitative phase of decision process. The same comment applies to other Group Decision Support Systems – like MEDIATOR (Jarke, 1987) or Co–oP (Bui, 1986). Therefore, supporting the qualitative aspects of decision processes needs more careful analysis.

This paper investigates the information processing aspects of SCDAS methodology. In fact, the following issues should be addressed:

- *Documents* existing in the system, their properties, rules for distribution, computer support for management and manipulation of such documents,

- *Communication* related aspects of organization of data sharing, location of data in server and workstation computers, synchronization of the process in time, communication and data exchange protocols etc.

This paper addresses only the first issue. This decision was motivated by the fact that the structure and general properties of documents must be specified before we start to investigate how to exchange these documents within the electronic network.

2 Documents structuring in SCDAS system

As it was mentioned above, most existing GDSS is oriented toward processing numeric information. The user of GDSS can enter numerical information to the system, retrieve this information, share with other users and perform rather complicated numerical procedures to extract important conclusions from this data. However, other types of information are also important for supporting decision processes. As it has been pointed out by Huber (1984):

> *"...Today's DSS are largely concerned with the retrieval and use of numeric information. In contrast, the environment of most meetings in corporation and public agencies is highly verbal. Thoughts are primarily shared and modified, not numbers. To the extent that the thoughts need to be recorded, they are put into text form... Meetings are extremely verbal environments, and the most important thoughts with which they deal are put into text form. A GDSS that does not reflect these facts will serve only a fraction of group tasks. For this reason it is important to consider how GDSS can support decision groups by aiding in the sharing of textual information..."*

As it has been pointed out in previous publications (Lewandowski, 1987), the most promising framework for implementation the group decision support system is the distributed computer environment equipped in teleconferencing and office automation software. Such an environment allows smooth transition from the existing practice of office and telecommunication systems utilization to new forms; moreover such an environment requires only small modifications or extensions to the existing software and methodologies to support new functions.

The basic idea of implementing the SCDAS in teleconferencing framework is the *extension of the concept of document.* In the standard office automation and teleconferencing systems the *text* - letters, memoranda etc. constitute the basic information carrier. In the *extended* or *decision* teleconferencing system the concept of *document* has been generalized - besides of textual data, numbers are transmitted between the members of the group. Moreover, the *formalized knowledge* necessary to interpret the data and to structure properly the decision process must be implemented within the system and made available in sufficiently simple and friendly form to the users of the system. Therefore, several types of documents can exist simultaneously in the extended teleconference system - documents which can be different nature and strongly interdependent. These dependencies can reflect logical relationships between numeric and textual data as well as can reflect the users opinion and knowledge related to the information being processed.

In order to design such extended teleconferencing system it is necessary to specify the possible types of documents generated and processed both by users and the system and rules for generating and processing of such documents.

Several types of documents can be distributed during a typical meeting. According to the agenda of SCDAS decision conference, some steps of the decision process can be more oriented towards quantitative reasoning based on analysis of numerical or quantitative data, whereas the other require strong exchange of verbal information, a lot of discussion and more qualitative oriented analysis. The insight into the consecutive steps of the SCDAS process leads to the following conclusions:

The first stage. In the existing experimental implementations of the SCDAS system (Lewandowski, 1987) it was assumed, that the SCDAS conference begins with *Phase 0*, when all elements of the decision problem (i.e. *alternatives, attributes, committee members* and a *procedure*) are known; therefore during the *Phase 0* all these informations can be entered into computer. In the fact, reaching such a high level of common understanding and consensus within a committee could require a lot of discussions and information exchange. Neither the list of alternatives, nor their descriptions need be complete at this stage; moreover, this information might be not known to the committee members at this stage, if they wish to avoid the bias in specifying attributes and their aspiration levels. The important issue at this stage that requires discussion and specification by the entire committee is the definition of the attributes of the decision and their scales of assessment.

Usually, there will be not too much doubts and discussions relating to the list of committee members; similarly, the set of alternatives will be usually given *a'priori* or prepared by experts. More complicated issue is the problem of attributes - what attributes are relevant to the problem being solved, what measures will be used to express the value of attributes, whether the set of attributes will be splited into subsets and aggregation will be performed etc. This discussion is extremely important - if committee members will have no common understanding what, for example is the meaning of "good" for the attribute "scientific reputation" - the whole process will lead to rather meaningless results.

Similarly, the procedural organization of the decision process needs intensive discussion - the questions which can be addressed on this stage of the process can relate for example, to the problem of recourse in decision process if some participants want to repeat some parts of discussion, to the question of organization of the process if some committee members are rejected from the committee or join the committee, to the question of organization of the meeting in time and coping with deadlines, etc.

The questions formulated during this stage of the discussion could include the following:

1. What is the expected product of the committee work and how does it influence the selection of the details of the procedure? The answer to this question depends on the committee's charter and its perceived role. For example, if the expected product is a short list of significantly different alternatives, procedural rules will be different from the case when the expected product is a consensus opinion on one, "best" alternative.

2. What rules for aggregating opinions across the committee should be adopted, in particular, should outlying opinions be included in or excluded from aggregation?

3. Should the committee be allowed to divide and form coalitions that might present separate assessments of aspirations, attribute scores and thus final rankings of alternatives?

The other phases of the decision process are much more precisely defined - it is clear what numerical (or qualitative) data is required on every step of the process and what formal operations should be performed with this data. There are however still several aspects of every phase of the SCDAS process which require discussion between the committee members. Let us analyse the consecutive phases of SCDAS process.

The second stage of the the decision process is devoted to *aspirations*. During this phase aspiration and/or reservation levels for all attributes are determined separately by each committee member. After these values are entered into the decision support system, all necessary indicators (disagreement indicators, dominant weighting factors - see further comments) can be computed. During this phase there will be no active information exchange between committee members - everybody should analyse the problem and specify aspirations himself.

The third stage has again two objectives. One is the analysis and discussion of aspirations by the entire committee. These discussions are supported by the computed indicators and their graphic interpretations. In these discussions, the committee might address the following questions:

1. Do the computed indicators accurately reflect the perceptions of individual committee members about the relative importance of various attributes (if not, should the aspirations or reservations be corrected)?

2. What are the relevant differences of opinions between committee members and do they represent an essential disagreement about decision principles?

3. Does the entire committee agree to use joint, aggregated aspirations (reservations), or will there be several separate sub-group aggregations?

The second objective of the third stage is a survey of alternatives. Discussions might centre on the following issues:

1. Are the available descriptions of alternatives adequate for judging them according to the accepted list of attributes? If the answer is negative, additional information should be gathered by sending out questionnaires, consulting experts etc.

2. Which of the available alternatives are irrelevant and should be deleted from the list? Such preliminary screening can be done in various ways. The committee might define some screening attributes and reservation levels for them (of a quantitative or simple logical structure): for example, we do not accept investments which are more expensive than a given limit.

The fourth stage of the decision process is the individual assessment of alternatives. The evaluation of each attribute for each alternative is the main input of committee members into the system. Each member specifies evaluation scores; the decision support system helps him by displaying the evaluations already made and those still to be entered.

When all evaluations are entered, a committee member should proceed to the individual analysis of alternatives, based on calculations of an achievement function that leads to a ranking of all alternatives for the given committee member. This ranking is the main source of learning about the distribution of alternatives relative to aspirations.

The questions addressed by each member at this point might be as follows:

1. Do the rankings along each attribute correctly represent the individual's evaluations of alternatives; does the achievement ranking, based on individual aspirations, correctly represent the aggregate evaluation (if not, should the scores be modified)?

2. If the committee member agrees with the individual achievement ranking proposed by the system, what are the differences between this ranking and that based on individual scores but related to committee aggregated aspirations? Are these differences significant, or can he accept them as the result of agreement on joint decision principles?

The fifth stage of the decision process relates to an aggregation of evaluations and rankings across the committee and consists of a discussion of essential differences in evaluations, followed by a discussion of disagreements about a preliminary ranking of alternatives aggregated across the committee. These discussions are supported by the system; the system computes indicators of differences of opinion and prepares a preliminary aggregated ranking.

The questions addressed by the committee at this point might be the following:

1. On which attributes and alternatives the largest differences in evaluations between committee members are observed? Do these disagreements represent essential differences in information about the same alternative?
2. What is the essential information (or uncertainty about such information) that causes such disagreements? Should additional information be gathered, or can certain committee members supply this information?
3. Would the results of these discussions and possible changes of evaluations influence the preliminary aggregated ranking list proposed by the system? This can be tested by applying simple sensitivity analysis tools.
4. Does the preliminary ranking proposed by the system correctly represent prevalent committee preferences?

After these discussions, a return to any previous stage of the process is possible. If the committee decides that the decision problem has been sufficiently clarified, it can proceed conclude the fifth stage by the final agreement on the aggregated ranking or selection of one or more alternatives. It is important to stress again that the committee needs not stick to the ranking proposed by the system, since the purpose of this ranking - as well as of all information presented by the decision support system - is to clarify the decision situation rather than to prescribe the action that should be taken by the committee.

It follows from the above, that committee members should perform rather careful and deep logical analysis of the decision situation - analysis of relationships between the set of objective data, their own opinions, opinion of committee confronted with their own expectations, aspirations and possibly biases created by factors not directly incorporated in the theory backgrounding the decision support system. Performing such a logical analysis can be a quite complicated task. Big amount of numerical data, complex relationships between data, possible problems in interpreting results generated by the computer and large amount of textual information constitute the basic factors making this analysis difficult. Therefore, in order to simplify the analysis certain support in *interpreting* the information should be provided by the computer. The simplest way to achieve this goal is introducing methods and tools for structuring this information.

In the principle, there are two possible strategies for handling large amount of information in order to simplify the analysis of this information by a human: *filtering* and *structuring*. The *filtering* strategy has been successfully used in many practical information systems (e.q. LENS, see Malone, 1987) and bases on the set of filters (rules) predefined by the system designers and possibly augmented by the user. These filters, usually build in the form of rules which can initiate some actions (triggers) when satisfied. Usually the user has full freedom in extending the set of rules and triggers.

In the *structuring* paradigm, the separate units or information are linked together, where some links can be predefined by the system designer and some can be defined by the user. All hypertext systems utilize this way of information structuring (Conklin, 1987).

The standard teleconferencing systems introduce certain level of information structuring. In the simplest case the conference has a tree-like structure: *the conference* is splited into *discussions*, discussions are splited into *topics*. Conference participants simply add their comments into the common pool of documents organized as linked list. Other types of conferences can exist with different structuring principles.

With information structuring offered by standard teleconferencing systems, the conference participant can have certain difficulties with analysis of information generated and distributed within the computerized mail or teleconferencing system - frequently it can be difficult for him to find important notices in the information flooding every day his computer. The issue of *informational overload* has been studied in details by Hiltz and Turoff (1985). They stated that:

> *"...The volume and pace of information can become overwhelming, especially since messages are not necessarily sequential and multiple topics threads are common, resulting in information overload... Unless computer-mediated communication systems are structured, users will be overloaded with information. But structure should be imposed by individuals and user groups according to their needs and abilities, rather than through general software features..."*

The term *electronic junk* introduced by Denning (1982) reflects well the situation which can exist in computerized document exchange systems.

To overcome this problem, several attempts to introduce some level of organization and structuring in computerized message system have been made. In the principle, two approaches can be applied for this purpose:

1. all formalized knowledge about the problem being supported is embedded into the system. The electronic form processing, calendar management system etc. belong to this category. The typical representative of this approach is the ODYSSEY knowledge-base assistant for travel planning (Fikes, 1981).

2. tools for defining logical relationships between documents are incorporated into the system, abstracting from the specific features of the problem being supported. All hypertext systems belong to this category, with INTERMEDIA as the typical representative of this approach (Yankelovich, 1985, 1988).

Recently, new concept of structuring information in teleconferencing systems have been introduced by Malone at al. (1987). He introduced the concept of *semistructured messages*. According to Malone:

> *"... semistructured message is a message of identifiable types, with each type containing a known set of fields, but with some fields containing unstructured text or other information..."*

Malone points out several reasons, why the idea of *semistructured messages* can be important:

- Semistructured messages enable computers to process automatically a wider range of information than would otherwise be possible,

- Semistructured messages allow people to communicate nonroutine information without the constraints of a rigid structure,

- Much of the processing people already do reflects a set of semistructured message types,

- Even if no automatic processing of messages were involved, providing a set of semistructured message templates to the authors of messages would often be helpful,
- Semistructured messages simplify the design of systems that can be incrementally enhanced and adopted.

In addition to the general reasons mentioned above, the concept of semistructured messages simplifies system design by:

- arranging the message types in a frame inheritance network so that specific message types can *inherit* properties from more general types,
- using a consistent set of display-oriented editors for composing messages, constructing message processing rules and defining new message templates.

The concept of semistructured messages have been utilized in the LENS system developed by Malone at all. (1987). He presents also several applications of this idea - in the field of teleconferencing, task management, calendar management etc.

An idea of structuring information similar to the semistructuring concept has been introduced by Cook at all. (1987). He analyses the possible taxonomies of information in the context of computerized support systems for project management (system NICK). However the purpose and the background of this system is a bit different than SCDAS, the introduced taxonomy fits well to SCDAS framework. Cook categorizes the information using two attributes: *ownership* (private, subgroup and public) as well as *structural properties* (binary, structured and public). The structured information has regular form and can be easily interpreted by a computer. This information might include lists, matrices, vectors, templates etc. Unstructured information is information that is not in regular form, like text and bitmaps. See Cook at all. (1987) for further discussion.

Why the idea of semistructured messages can be important for SCDAS implementation? To investigate the problem we should analyse first the possible types of information which can be processed within the SCDAS system. This information can be categorized according to two attributes: *information access and ownership* as well as *structural properties* of information.

The *access to the information* generated during the SCDAS session depends on two factors:

- *the privileges of the individuals participating* in the SCDAS conference. The rules are simple - the *conference owner* (or *committee president*) is the only person authorized to change the definition of the problem - like adding new committee members or removing them, changing the list of attributes or list of alternatives etc. He also can generate the textual informations relating to the problem definition or to the progress of the conference, which have *read-only status* for other committee members. Moreover, he can decide whether at a given stage of the process this information can be visible to other conference participants or will be hidden.
- *the stage of the process.* Since the SCDAS conference has some temporal dimension - the decision process advances from the given stage to the next one if all committee members specified all information necessary on the given stage, the access rules can change in time.

With respect to structural properties, the information generated during the SCDAS conference can belong to two classes:

- the highly structured *numerical and qualitative data*. All the information constituting the problem definition, the relating information generated by the participants (aspirations, scores) as well as information generated by computer (values of achievement function, rankings, graph plots, etc.). There exist strong and well defined relationships between these data - we will call these relationships *structural links* in this sense that it is well defined what data are required from the conference participants at a given stage, what properties these data should possess, what actions (and calculations) are necessary to perform when data are entered to the system or changed by the user and what data must be used to calculate other numerical information.

- the unstructured *textual information* - like notes, memoranda, mail notes send to other conference participants. This information is similar to these generated and distributed during the standard conference. The only difference is in the structuring principle - usually, some part of this set of information can strongly relate to the numerical data. Therefore, the numerical data can be treated as the equivalent of *topic* in the standard conference - for every numerical item there can exist the linear list of comments generated by conference participants. Therefore the *hard links* between textual documents can exist - two linear links of comments will be *interrelated* if there exist some links between numerical data which this textual information is associated. We will call these links *hard* since they are *a'priori* determined by the organization of SCDAS procedure.

Summarizing, the information generated during the SCDAS conference can be structured by the *structure of the decision process* itself. It is possible, however, that the second layer of links between numerical data and textual information can exist - namely links introduced by the user in order to reflect his particular, personal view on various aspects of the problem being solved. This kind of relation between documents we will call *soft links*.

The *soft links* can be arranged in similar way like it is done in hypertext system. In this way we have two, parallel layers of links - the *soft layer* and the *hard layer*. Therefore in the contrary to the standard hypertext we will have the *primary relevant documents* and the *secondary relevant documents* - depending on the fact whether relevant documents are belonging to the same layer where the root of the search tree is located.

It is necessary to mention, that the *soft layer* can be further splited into sublayers - the *public sublayer* and the *private sublayer*. The public links can be generated either by the committee president or by authorized conference participants. The private links are known only to the user who is creating them and constitute the part of his local information base (*the notebook*).

The difference between hard and soft links are not only formal - their existence can support different questions relating to the problem being analysed. If the question addresses the problem *"...how to explain the fact that my favourite alternative is ranked by the committee so low..."* in order to answer it is necessary to know what data directly influence this fact. This can be difficult issue for the user, especially if he does not know exactly the theory backgrounding the system. Since this theory is known to the system developer, he can establish the links which can help to trace data which are relevant to the posed questions. It is clear, that the answer on such question depends on the *state of the system* - understood as the values of data present in the system on the given stage of decision process. Therefore, some of these links can be *dynamic*, i.e. they can be changed during the progress of the decision process. Therefore, the mechanism for creating an updating such links must be built into the system. In order to fully support this function of the system, hard links must be provided for *help documents*. These documents play the role of standard context dependent help, but similarly like dynamic links these documents can also be dynamic - the help given to the user must depend not only the current state of the program (i.e. to address the question *where am I now*) but also address the current state of the system (*what follows from this*). These two functions of the system we will call the ***guidance help*** and the ***explanatory help***.

Evidently, some questions formulated by conference participants do not require dynamic links. If the user, specifying scores for alternatives wants to know everything about a given attribute, he can be guided through documents relating to analysis of aspirations concerning this attribute, definition of this attribute etc., independently on the state of the system.

The soft links play role of the ***remainder*** - the user can link and browse documents which he, or other participants consider as important on a given stage of the process. Evidently, these documents can contain both the numeric and textual data.

Similar categorization of links has been made by Conklin (1987) for hypertext systems. He distinguishes between ***referential links*** and ***organizational links***. According to Conklin, the referential links usually act as reference. The "destination" of the link usually functions as the referent - the material which due to some reasons is relevant to the material located at the beginning of the link. The organizational links correspond to the logical structure of the information. In SCDAS, the variety of possible links is however bigger, but the organizational links can be considered as the analogy to hard links, whereas the referential links - to the soft ones.

It is necessary to point out the difference between the hypertext system, like INTERMEDIA and the "hypertext" concept used in SCDAS. Namely, in the standard hypertext links can be established between words, sentences or paragraphs of a given document and other documents. It seems, that for applications like SCDAS and teleconferencing this level of detail is not necessary. Therefore we will assume that links can be created only between full documents and data. Moreover, the concept of ***the node*** in information structure of SCDAS system is much more complicated than in a standard hypertext. On the top level of hierarchy we have ***data nodes*** – the information structures responsible for storage and manipulation of certain class of data. From the user point of view these nodes represent the ***active electronic forms*** to be filled by the user. Similar concept of ***semistructured nodes*** has been explored by Conklin and applied in ISAAC hypertext-like system for supporting software design process.

Summarizing, we can view the SCDAS decision conference as the document exchange problem with documents being ***procedurally structured*** and ***contextually structured***. Therefore this concept goes beyond the idea of semistructured messages implemented in LENS system - due to the strong logical backgrounds of SCDAS the level and complexity of structuring can be much higher.

3 Structured documents in SCDAS system

Let us discuss in details the structured documents which can be generated during the SCDAS session and possible relationships between them. The structured documents can be generally categorized into numerical and textual ones. As it was mentioned in previous sections the structured documents can be *generated by the conference participant* or *computed by the system*. The questions formulated now can be as follows:

- what types of structured data is required from the user on a given stage of the decision process and what the operational rules for handling these data (e.q. how to handle missing data, verify the correctness of data etc.),
- how these data can be used by the system on a given stage of the process,

- what are the possible dependencies between data entered by the user and/or generated by the system on various stages of the procedure.
- what actions are undertaken when a given action related to data is performed by the user.

The structured information required from the user (users) depends on two factors:

- the current phase of the SCADS process,
- the privileges of the user entering and manipulating data.

The procedural framework presented in paper by Lewandowski and Wierzbicki (1987) specifies all the data created during performing the decision-oriented part of the conference. The data can be split into three following groups:

- Data characterizing the problem being solved. These data are generated by the conference owner and contain all the information necessary to initiate the conference. They include:
    1. List of *alternatives*, together with all documents characterizing these alternatives and necessary to make evaluation,
    2. List of *committee members*, together with *voting power*, specifying number of votes assigned to each committee member,
    3. List of *attributes* together with *numerical or verbal scale* necessary to express the value of attributes together with all relevant documents concerning the given attribute. Except of full name of every alternative, committee member or attribute, the system utilizes 3 character long abbreviations which can be used for graphic presentation, information display and internal coding.
- Data created by the user during interaction with the problem. These data include:
    1. Values of *aspiration and reservation levels* specified for each attribute,
    2. Values of *scores* for all alternatives reflecting the subjective *value of alternative* with respect to all attributes,
- Data generated by the system during iteration process. They include:
    1. *Average aspirations and reservation levels* for all attributes,
    2. Values of *achievement functions* computed for scores specified by all committee members and for individual as well as committee aspirations. These functions are used by the system for ranking alternatives. See paper by Lewandowski and Wierzbicki (1987) for formulas and procedural details,
    3. *Ranking data* which reflect the ordering of alternatives according to the information specified by conference participant *individual aspiration* and *individual scores* as well as information relevant to the committee opinion *aggregated aspiration*,
    4. *Status indicator* generated by the system as the response for user's actions. Every user has his *local status indicator*, the system computes the *global status indicator*. These indicators reflect the phase of decision process and are equal to the number

of phase being currently processed. The system compares individual status indicator with the global one and on the basis of this information determines what data are accessible for the user and what actions he can undertake. When the user terminates the current phase, his local indicator is incremented; the global indicator is incremented only if all users successfully completed the current phase. The global status indicator can be manipulated by the conference owner - he can, for instance *decrement* this indicator to make the recourse in decision process.

The rules specifying access to data created and analyzed during the decision conference are as follows:

- The *conference owner* has access *to all data* created during the conference with the following access rights:

  1. He is the only person authorized to change the problem definition, i.e. list of committee members, attributes and alternatives *(read - write access)*,
  2. He has access to all data generated by the users, i.e. aspiration and reservation levels, scores assigned to alternatives and user's status indicators or computed by the system using user's data, like average aspirations or user's achievement functions*(read-only access)*,
  3. He can change the status indicator; this action will allow changes of user's status indicators *(read-write access)*,

- The *conference participant* has access to the following data:

  1. All data defining the problem, i.e. list of committee members, attributes and alternatives *(read-only access)*,
  2. Data computed by the system, like average aspirations and global achievement functions *(read-only access)*,
  3. His own data like aspirations or scores assigned to alternatives *(read-write access*, or *read-only access* depending on the current value of status indicator*)*,
  4. All the data located in his own notebook *(read-write access)*
  5. Data created by other user can be accessible by other users only with permission of the conference owner *(read-only access)*.

Except of definition of data structures, we should investigate the temporal dependencies between data generated on various stages of decision making process as well as rules for sharing data between users during each stage of decision process. This analysis is important, since according to the procedural framework the decision making process evolves in time. Therefore the following aspects should be investigated:

- What data are generated at every stage of the decision making process,
- How the data access rights are changed during this process,

The rules for data access are relatively simple: on a given stage of SCDAS process the conference participant has free access to data generated by himself and by the system during previous stages. This access is however restricted and such data can be *only inspected.* On a given stage of the process the system requests from the user some data; he can freely modify and read this data until he decides to terminate the current phase. It happens, when the conference participant decides that entered data reflect well his point of view about the problem and can be used for further computations. Since this moment the data is *locked* and available only for reading and inspection. Termination of the session changes the *local status indicator* (see above) which is used by the system for synchronization control. The committee president can change the status indicator what results in unlocking the data generated during previous stages. In this way the recourse in decision process can be performed.

Let us concentrate on details of operations performed during every step of decision making process:

- *Phase 0.* During this phase the conference president can initiate the new conference of update the old one. The standard sequence of actions undertaken during this step consists of the following:

  1. Specification of user's name and verification of access mode (the ordinary conference participant or the conference president)
  2. Verification of the user's name. If such a name is not known to the system, new conference should be initiated.

  Exit from this stage of the program is possible in two modes - the *quit* mode and *terminate mode*. In *quit* mode the program terminates, but the data are not transferred to the global data base. Therefore, the user can invoke the program again and perform necessary data modification. It *terminate* mode, all data are transferred to the data base. In this case the *local status indicator* is also updated as well as the *global status indicator*.

- *Phase 1.* In this phase all users should define aspiration levels for all attributes. The preamble of this phase is similar to the previous one:

  1. specification and verification of user name. The request is rejected if the specified name is not known to the system (i.e. he is authorized to participate in any conference already defined). The request is also rejected if the user terminated the current phase and wants to modify some data - according to the procedure it is not possible without acceptance of the conference president.
  2. creating (or updating) the data base with the list of attributes and numerical data associated with attributes. Similarly like in previous phase, the program can be terminated in two modes - the *terminate mode* quits the program only; the data are not transferred to the data base and status indicator is not updated. Therefore the user can resume this phase as many times as he requires until he decides that he specified all required data. In such a case he should exit with a *quit* mode, what initiates updating the global data base with new aspiration data. The system checks the status indicators of all users - if all of them completed the current phase, the global status indicator is incremented. In such a case the users can begin performing the next phase of the process.

- *Phase 2.* In this phase the user can perform the analysis of data specified during the previous phase. After verifying the user's name and the status indicator, the user can perform all necessary data analysis. Similarly like in the previous phase, it is possible to exit the workstation program in *terminate* or *quit* mode. The standard procedure for incrementing the status indicator is performed.

- *Phase 3.* In this phase the user must specify scores for all attributes. This is definitely the most complicated and time consuming phase of the decision process which may request rather intensive interaction with other data information systems and services available to the user as well as rather deep analysis of the specified data. After terminating this phase the score table is transferred to the global data base and the standard procedure for incrementing the status indicators is performed.

- *Phase 4.* In this phase the final analysis of the data is performed. When all users completed the previous phase, the *achievement functions* for scores specified by all committee members are computed (see Lewandowski and Wierzbicki, 1987 for details and formulas). Values of these functions are used for ranking alternatives. Since this is the last phase of the process, status indicators are not updated.

4 Unstructured documents in SCDAS system

As it was mentioned in previous sections, except of highly structured information related to alternatives, attributes, scores etc., the SCDAS system supports generation, exchange and analysis of several types of *unstructured* information. This function of the system supports the *soft* side of the decision process – in many cases more important for obtaining final result and usually requiring more effort to complete than just collection of scores and computing of rankings.

The basic element of this side of the process is *the document.* We will understand this term in narrow sense – the document will be *non-active, textual or graphic information.* We will not consider more general case, when the document can be *active* – i.e. distributed together with tools for analysis of this document. This is not necessary, since all data analysis in SCDAS is concentrated in data nodes of the information structure; if the conference participant wants to perform some *what-if* analysis he can easily duplicate the data node and make his private copy; all tools for analysis are available to him all the time (with some natural constraints following from the SCDAS procedure).

Each document can belong to one of four groups of documents:

- *Public documents* which are generated by the committee president. These documents contain general information about the particular aspects of the problem – in the case of attribute it can be, for example, detailed explanation of the meaning of this attribute, in the case of alternative – information about this particular alternative, like curriculum vitae for personnel selection problem, details of the project for project evaluation problem etc. To the sane category belong *help documents* constituting the part of system implementation. It is possible however, to extend the set of help documents or change their content in order to create customized version of the system to fit to the level of knowledge of the audience participating in the SCDAS conference. This is important, since different levels of help are necessary for such different problems like software project evaluation or personnel selection.

- *Message documents* which are generated by conference participants as their contribution to the discussion. These documents are available to all conference participants, however the only person which can modify these documents or remove them from the system is the conference president.

- *Private documents (notes)* which contain the information generated by the conference participant and stored for himself for future utilization *(the notebook).* Any document available to the conference participant can be imported to the notebook, including help, public and message documents. Moreover, any information available in data nodes can be imported to the notebook – with this restriction that whereas in the data node information is *active*, i.e. can be processed by tools implemented in the system, the information imported to notebook is *passive* – i.e. treated only as sequence of characters. As it was mentioned above, this assumption does not restrict the flexibility, since every time the user can create his own instance of the problem to make himself the analysis. Such a copy can be also treated as a part of the notebook.

- ***Mail documents*** which contain information received from other conference participants or send to other participants. This information is accessible exclusively for a person identified as the receiver and the author.

The documents created during SCDAS conference are *linked*. As it was mentioned in the previous sections, some links have ***organizational*** character – i.e. they are predefined by the SCDAS procedure. Documents can be also linked by ***referential links*** pointing the information which not necessary belongs to a given category, but can be interesting or relevant from a given point of view. Usually, these links do not reflect ***the logical relationships*** between data, but rather ***the contextual*** relationships. All conference participants have full freedom to create referential links – both between structured and unstructured documents.

The referential links are private property of the conference participant – they constitute the part of his notebook and he is responsible for their creation and deletion. The only exception relates to links pointing to public documents – since these documents can be removed by the conference president, the corresponding links can be deleted by the system automatically if the document is not imported to the notebook.

Some referential links can be created by the conference president; they are known to all conference participants and clearly distinguished from the private ones (***physically*** only, i.e. using different screen forms; logically they not differ from private links).

Like in standard hypertext systems, several problems related to linked information structures can appear. These problems are discussed in details by Conklin (1987) and we will not repeat this discussion here. We will just mention two possible problems: ***disorientation*** and ***cognitive overhead***. The first problem relates to large amount of information which can be generated and manipulated by the system – the user can be easily "lost in space" – lose his sense of location and direction in a linked document. The second one related to the additional mental overhead required to create, name and keep track of links. Several level of concentration is necessary to maintain several tasks at one time. It seems however, that in SCDAS system the mentioned above problems do not constitute a big danger – due to high level of structuralization most links and tasks related to information linking are either predefined or not a'priori defined but easy to understand for someone knowing the SCDAS procedure. Several measures relating to ergonomic aspect of screen and user interface design must be taken; the problems appearing here are the sane like in hypertext and will be not discussed here.

5 Components of the system

The important problem arises when defining the software system like SCDAS – what specification methodology should be used to define functions of the system. The problem has been analysed from many points of view and several technologies for program specification have been formulated. The review of several techniques can be found in book by Schneider (1979);

the very general and formal approach has been proposed by Liskov (1986). One of the most advanced and formal tools for specification is ALPHARD (Shaw, 1981).

Recently, instead of procedural or functional paradigm (see, for example de Marco, 1979) the object oriented paradigm has been introduced. Again, we will not discuss here the principle ideas of object oriented approach or make comparisons with other formalisms (see, for example, Booch, 1986 for more detailed discussion). We will emphasise only the major steps in object oriented development process. The most important issue is the definition of object. There exist some formal approaches for defining objects; for our purpose it is enough to give a verbal definition: the object is an entity with following properties:

- has state,
- is characterized by the actions that it suffers and that is required by other objects,
- is an instance of some class,
- is denoted by name,
- has restricted visibility of and by other objects,
- may be viewed either by its specification or by its implementation.

More deep analysis of this concept has been performed recently by Cox (1986); he investigates the impact of this programming paradigm to the software development process. The other important implications of object oriented development style have been presented by Yonezawa at all., (1987).

The main steps in software system design process base on the object oriented approach are as follows:

- Identify the objects and their attributes,
- Identify the operations suffered by and required of each object,
- Establish the visibility of each object with respect to other objects,
- Establish the interface to each object,
- Implement each object.

It is important to note, that several programming tools directly support the object oriented paradigm – some of them, like Smalltalk (Goldberg, 1983) or Trellis/Owl (Schaffert at all., 1986) implement this concept directly. Therefore, the specification based on such software tools can be directly executable. It does not mean, that this approach requires some specific tools – for example the ADA language, not directly designed with object oriented paradigm as a design assumption can be successfully used for this purpose (Booch, 1986, Buzzard, 1985). Even PASCAL, the language very far from object oriented paradigm can be applied for object oriented design of software systems (Jacky, 1986, 1987).

The object oriented approach seems to be the best framework for designing and specifying the SCDAS. The system consists of several objects interacting during decision process: committee members, alternatives, attributes, rankings, documents. The number of existing objects can change in time due to progress in decision process, relationships between objects can also change in time. Therefore, using the object oriented paradigm it is relatively easy to specify components of the system and interaction between these components.

All objects existing in SCDAS system can belong to two categories: ***passive objects*** and ***active objects***. Instances of passive objects can be defined and manipulated by conference participants or by computer. These objects contain data constituting problem definition, information specified by conference participants, values of achievement functions computed by the system, ranks, graph plots etc. In other words, passive objects are ***data structures*** holding information being processed by the system. Active objects constitute a bridge between passive object, conference participants and computer. They can be used for data definition, data inspection, data analysis, controlling the computation process, controlling the interaction between users, provide tools for data analysis etc. In the fact, active objects constitute the ***user interface*** in the broad understanding of this term – or the ***organizational interface*** according to Malone (1987).

It is necessary however to make the following remarks:

- there no exist good and commonly acceptable standards for object specification, especially for specification of operations (methods according to commonly used therminology). Therefore we will use the verbal description only to define the attributes of the object and inheritance; the operations can be defined in terms of SMALLTALK source code. Due to lack of space and the purpose of this paper these operations will be not discussed here; a separate publication will adderss these issues.
- the system is currently being specified and not all objects and operations are completely defined.

5.1 Passive components

As it was mentioned above, ***passive components*** of SCDAS system are data structures containing all information relevant to the problem being analysed. The specification of passive data classes includes:

- specification of ***private variables*** which contain all information characteristic to this particular class,
- creation of ***instantation protocol*** which causes creation of instance of a given class,
- specification of ***access protocol*** which allows access to local data elements from the outside of the class instance.

In this way all passive elements can be treated as encapsulated containers with information relevant to small part of the problem definition; these data are accessible from the outside by other objects when necessary. A good metaphor of object oriented paradigm was given by Budd (1987):

> ***"..Instead of a bit-grinding processor raping and plundering data structures, we have a universe of well-behaved objects that courteously ask each other to carry out their various desires..."***

In the sequel we will discuss the passive components of SCDAS system.

5.1.1 Basic data

The basic data contain all information about the problem being solved, i.e. about ***committee members, attributes*** and ***alternatives***. Since there exist certain similarities between classes responsible for information storage for all three components, the basic data classes are organized as hierarchy. On the top of hierarchy the class *DataInfo* is defined. This class posses the following private data elements:

NAME This is the name associated with the instance of the class, i.e. name of the committee member, the attribute or the alternative,

ABBREVIATION This is the abbreviation (not longer than 4 characters) which is used by the system as a key for data access as well as for referring to a given data instance during graphic presentation and display of results,

NAME RELEVANT DOCUMENTS This attribute contains all general documents relevant to the particular data instance. According to the terminology introduced previously, these documents belong to the category ***instance relevant documents*** and can contain such data like information about alternative (e.q. CV of a candidate in personal selection problem, details of the project, information about technologies considered during decision process), explanation of the meaning of the attribute or telephone number and other informations relating to the committee member. The class ***document*** will be discussed later; here we will mention only, that in the fact the attribute ***document*** consists of four subgroups – ***public documents***, ***message documents***, ***note documents*** and ***mail documents***.

a. Committee member

The object *committee member* provides the data container for all information generated by the user and required by him to perform analysis of the problem or to communicate with other committee members. This object possess the following attributes:

VOTING POWER This is the number of votes assigned to the committee member; this number is used for calculating averages and achievement functions (to avoid misunderstandings, it is necessary to point out, that SCDAS does not directly utilizes voting procedure. See paper by Lewandowski and Wierzbicki for more detailed explanation),

ASPIRATION TABLE This is the table containing aspiration (reservation) values specified by the committee member for all attributes,

SCORE TABLE This table contains the scores assigned by the committee member to all alternatives and attributes,

MAIL BOX The mail box contains all messages send to other committee members or received from other committee members,

NOTEBOOK The notebook contains all documents which the committee member considered as relevant or important from the point of view of problem analysis,

It is important to note that most of the attributes of the defined above object are *also objects*. These objects will be defined in following sections of the paper.

b. Attribute

This object contains all information about the *attribute*. Similarly, as the object *committee member*, this object has name and abbreviation. Moreover, since the SCDAS methodology assumes that the "value" of the attribute is expressed in numerical scale, for each attribute the *maximum* and the *minimum value* must be specified. Moreover, there are documents associated with all attributes. These documents can be associated with attribute name, attribute maximum level or attribute minimum level. Summarizing, the object *attribute* has the following own attributes:

LOWER LIMIT This attribute contains the value of lower limit of the attribute,

UPPER LIMIT This attribute contains the value of upper limit of the attribute,

LOWER LIMIT RELEVANT DOCUMENTS This attribute contains all documents relevant to the lower limit of the attribute,

UPPER LIMIT RELEVANT DOCUMENTS This attribute contains all documents relevant to the upper limit of the attribute.

c. Alternative

This object contains all information about the alternative. This data are the simplest than the previously defined components of the system description. Since all alternatives are defined by experts outside of the system, the only information stored in a computer is the name, abbreviation and documents. Therefore, the attributes of this object are as follows:

ALTERNATIVE RELEVANT DOCUMENTS These documents contain all informations about the set of alternatives being analysed by the committee – like biography of a person applying for a position in the institute, details of a project proposed by Research and Development Department, details of a production technology. These documents constitute the most important part of problem definition and are located by the conference president in the *public* pool of documents. Any other comments and documents being the result of discussions are generated by conference members and located in the *message* pool of documents.

5.1.2 Dictionaries

The information structures relating to attributes, alternatives and committee members defined within one decision conference must be organized into *dictionaries*. The dictionaries are implemented using the standard *Dictionary* class of Smalltalk – the associative memory with content accessible through a key. In SCDAS the abbreviation is used as the key for information retrieval. Similarly like in the case of DataInfo class and its subclasses, the general class *DicInfo* has been defined; the subclasses *MemDic*, *AtrDic* and *AltDic* are responsible for maintaining information related to committee members, attributes and alternatives.

The only attributes of this class hierarchy are *documents*. Every dictionary has several documents associated with it. Contrary to documents defined in the previous section, these documents have more general character – they relate to the *concept* of a given component of the problem definition.
The following documents can be associated with the instance of *DicInfo* class:

HELP DOCUMENTS are generated by the system designer and they constitute the part of system implementation. These documents can be however extended by the conference president in order to fit their explanatory level to skills of other participants or to specific aspects of the problem being solved,

NAME DOCUMENTS contain all informations relating to the *class*, in contrary to documents defined in the *DataInfo* class which contain documents relating to a given, particular data instance. Therefore, according to the terminology introduced previously, the documents built around dictionaries and data structures constitute the logical hierarchy – with documents relating to the *class* on the top and documents relating to *instance* on the bottom. The relation between such documents is determined by their definition and follows from the relation between instances of class *DataInfo* and its subclasses and instances of class *DictInfo*. Therefore, links connecting these documents belong to the category of ***hard links***.

As it was mentioned above, elements of class *DicInfo* constitute similar hierarchy like basic data structures. The *DicInfo* class possesses certain general properties and general methods of this class; subclasses are responsible for managing information related to committee members, attributes and alternatives. It was also mentioned, that with every data instance there exist several associated documents – the ***instance relevant document***. Similarly, instances of *DicInfo* class posses also associated documents being the logical equivalents of corresponding documents of data instances. These documents addresses issues related to ***concepts*** like ***name of the attribute***, ***lower limit of the attribute***, ***upper limit of the attribute*** etc. Due to equivalence with data classes we will not discuss all these documents.

5.1.3 Documents

The object *Document* is the container for non-structured data like text or bitmap. The standard protocol for accessing and manipulating documents of different types has been defined, therefore all applications can use the same uniform interface to generate documents and to provide access to all documents existing within the system.

It was mentioned in previous sections, that in fact all documents existing in the system can be splited into four groups (five, if we consider also ***help*** documents. Therefore, every object "document" has hierarchical structure and is organized as *OrderedCollection* of *OrderedCollections* of unlimited size. These "low level" collections contain ordered list of *documents*.

This class is also organized as hierarchy, with abstract class *Document* on the top and classes *DocInfo* and *DocHelp* on the bottom. Such organization was necessary, since help documents require different type of cooperation with active components of the system, but despite of this fact part of access protocol is the same for both types of documents.

5.1.4 Nodes

As it was mentioned above, the user of SCDAS system can establish links between documents which he consideres important from the point of view of a given aspect of the problem analysis. These links constitute his own information base and are invisible for other conference participants. The *Node* element makes creation of such links possible.

Each conference participant owns the ***node*** associated with every document. The ***node*** is the instance of *OrderedCollection* class. Elements of this collection constitute all documents which conference participant considers relevant from the point of view of the problem being analysed. Since each document belonging to the *Node* list also owns its own *Node* object, all relevant documents can be organized in a complex data structure (graph). Clearly, such a graph schould not contain cycles (loops) since such a structure has not too much practical sense.

5.2 Active components

It was mentioned previously, that active components of the system play the role of ***interface*** – interface to the user, data bases, telecommunication system. In the sequel we will consider only first function of active elements – namely communication with the user.

There are several ergonomic aspects of building users interface; many of them have been discussed, for example by Schneiderman (1987). We will not analyse in this paper various aspects of this problem. We will mention only that the idea of interface follows the principles of SMALLTALK, with ***browsers*** and ***windows*** which allow direct data manipulation.

In the sequel we will discuss the active component of SCDAS system, but details related to private attributes and protocols will be not discussed. These components are conceptually more complicated than the passive ones since they must provide access to data and mechanisms for changing data. Therefore, active components must perform several functions like data checking and verification, filtering and verifying user's requests, analysing state of the system etc. Due to the complexity of these functions and the fact that organization of these components is system– and implementation dependent, details of their internal organization will be discussed in a separate paper.

5.2.1 Browsers

The user of the system needs flexible and easy tools for fast access to data, both structured and nonstructured which could allow easy data retrieval, analysis and modifications. These tools are called ***browsers***. Browser is in the fact the electronic form with the structure and organization reflecting the organization and properties of data which the particular browser should service. Since browser logically corresponds to the data node in the information structure of SCDAS, it must be equipped in necessary tools for executing transitions to other nodes and for supplying all necessary information relating to existing links. The following browsers can exist in the system:

- DATA BROWSERS corresponding to all possible types of data requested from the user. These include the *MemberBrowser*, the *AlternativeBrowser*, the *AttributeBrowser* and the *ScoreBrowser*. These browser have spreadsheet-like organization – they consist of several columns; each column corresponds to one component of the data item. All fields of the browser (including data fields) are ***sensitive***, i.e. the user can point such a field with a mouse to activate the corresponding menu. This menu is also ***state sensitive***, i.e. it can depend on the privileges of the user, the current phase of the SCDAS process as well as the actual values of data managed by the browser.

- DOCUMENT BROWSER. This browser allows reading, editing and distributing textual and graphical documents. Their general organization is similar to data browsers – all fields are sensitive and pointing to them activates the corresponding menu. They act also as active nodes and provide necessary mechanisms for traversing the data and documents network. Organization of document browser depends also on the type of documents processed – for example, different browser layout is necessary for processing mail documents and different for help documents. Except of the organizational and the selection windows, there exist the *editor* window for textual information displaying and processing with access to standard text editor. This editor includes the "copy" and "paste" functions which allow transferring some parts of text from one document window to the other (for example, public or message documents can be transferred to user's notebook). This window can be also used for displaying and editing graphical information; in this case separate tools for processing such an information are available to the user.

Two types of document browser exist in the system – the ***full document browser*** and the ***restricted document browser***. The first one can be activated only on the highest level of document tree, i.e. from the data browser. It can provide access to all types of documents associated with a particular data browser – like public documents, messages etc. If the user decides to traverse the information network, all other relevant documents will be displayed in the restricted browser with limited menu, allowing only traversing from one node to the other. Generally, documents displayed in the restricted browser cannot be changed.

- HELP BROWSER. This browser allows to read and create help documents related to a given topic. Help documents are organized as linked list of subitems, however, probably tree structure would be more adequate – see for example the hypertext-based help system for UNIX operating system developed by Brown (1986, 1988). Each subitem can be selected from the list of items displayed by the browser. Help documents can be accessed as *read only* for ordinary users; the conference owner can add new documents or delete existing ones. The standard help documents are provided by implementator of the system; these documents are loaded when new conference is initialized.

- NETWORK BROWSER. This browser allows to traverse the information structure of linked documents existing in the SCDAS system. In the principle, the user can move from one document to other related documents just selecting a proper item in the document browser menu. The *network browser* gives however more insight into structure of dependences between documents since it provides graphical presentation of links.

- PROBLEM BROWSER. This browser allows to select one particular *problem* among many problems which can exist simultaneously in the system. The user of the system can participate in many conferences; he can has the status of ordinary participant in some of them, or he can be the owner of other conferences. Therefore, the problem browser allows to have the preview of all conferences related to a given user. Depending on the status of the conference, the conference participant can use this browser to get access to available information according to his access privileges and status of the SCDAS process.

5.2.2 Windows

The *Window* are implemented in the system as the tool for active communication with the user. The following types of windows are available in the system:

- PROMPTERS. As it was mentioned above, browsers allow *reading* and *inspecting* the data and other information; if the user requests to make any changes in the data managed by the data browser, the window requesting such a change is created. This kind of windows we will call ***prompters***. Except of text editor, being the part of document window the prompter is the only available tool for entering data to the system or modifying the data. The prompter is ***context dependent*** in this sense that it provides control of correctness of the information required from the user depending on the current stage of the process and values of data. The ***text prompter*** requests information or data in the "standard" form – i.e. like string, text or number. The ***graphic prompter*** allows entering numerical or qualitative data using the pointing device.

- NOTIFIERS informs the user about results of actions, error conditions etc. The menu is associated with the notifier, which allows to undertake the action necessary to react properly in the occurrence of such conditions.

- DISPLAY WINDOWS are used for presenting results of numerical computations, as a sequence of numbers or graphically. This information is generated by the system and the user cannot make any changes. The only action which he can do is to select some part of the window and to import this information to his own notebook.

5.3 Problem

The *Problem* is the logical equivalent of ***discussion*** in computerized teleconference. According to the object–oriented specification terminology, ***problem*** constitutes the object possessing as private attributes all mentioned above components of decision problem definition. These attributes are as follows:

GLOBAL STATUS INDICATOR This attribute describes the current phase of the SCDAS process. As it was mentioned previously, all committee member has his own ***status indicator***; value of this attribute is incremented when the committee member terminates the current phase of the process. The local indicator controls access to local and global data of a particular committee member.

The ***global indicator*** is incremented when all committee members complete the current phase. This initiates all calculations (rankings, averages etc.) necessary to initiate the next phase. Moreover, the data access rules are changed according to general principles of SCDAS procedure.

MEMBER DICTIONARY This attribute contains all information about conference participants (committee members). When the object PROBLEM is instantiated, the new object MEMDIC (see sec. 5.1.2) is created and assigned to this attribute. The conference owner can use the corresponding browser to enter data about committee members or import these data from other existing instances of object PROBLEM.

ATTRIBUTE DICTIONARY This attribute contains all information about attributes (quality factors) necessary for evaluation of alternatives. Similarly, like in the previous case, instantation of the object PROBLEM causes creation of the new instance of the object ATRDIC; this instance is assigned to the attribute ATRDIC.

ALTERNATIVE DICTIONARY This attribute contains all information about alternatives. Similarly, like in the previous case, instantation of the object PROBLEM causes creation of the new instance of the object ALTDIC; this instance is assigned to this attribute.

MESSAGE DOCUMENTS This attribute contains all messages generated by conference participants during interaction with the system.

PUBLIC DOCUMENTS This attribute contains all documents generated by conference president and available to all participants.

As it was mentioned previously, the ***problem browser*** is associated with the class *Problem*. This is the main control mechanism allowing data creation and inspection, access to documents, etc. All data constituting the problem definition as well as defined by conference participants and calculated by the computer (alternatives, aspirations, scores, documents) are accessible through the ***problem browser*** and corresponding specialized browser invoked from the level of ***problem***.

6 Implementation

The ideas presented in the paper have been experimentally implemented on the IBM-PC computer and the Vax-Mate (the IBM-AT compatible manufactured by DEC) using the SMALLTALK/V programming language (DIGITALK, 1986). The main purposes of this implementation were as follows:

- to prototype the user's workstation for distributed group decision support system based on the SCDAS methodology and utilizing the standard teleconferencing software like the Telecenter developed at IIASA (Pearson at all., 1981, Fuhrmann, 1987) or the NOTES teleconferencing system manufactured by DEC,
- to investigate the feasibility and efficiency of presented concept of documents structuring,
- to clarify the ergonomic aspects of the user interface.

Details of design and implementation of prototype system as well as principles of cooperation between the workstation and teleconferencing software will be presented in a separate publication. In the Appendix we will briefly describe the functional aspects of the system in order to clarify the issues related to the mentioned above ergonomic aspects of user interface design.

7 References

Brown, P.J. (1986). Interactive Documentation. *Software-Practice and Experience*, Vol. 16, pp. 291-288, March 1986.

Brown, P.J. and M.T. Russell (1987). Converting Help Systems to Hypertext. *Software-Practice and Experience*, Vol. 18, pp. 163-165, February 1988.

Budd, T. (1987). A Little Smalltalk. Addison-Wesley Publishing Co., 1987.

Bui T.X. and M. Jarke (1986). Communications Design for Co-oP: A Group Decision Support System. *ACM Transations on Office Information Systems*, Vol. 4, No. 2, April 1986.

Buzzard, G.D. and T.N. Mudge (1985). Object-Based Computing and the Ada Programming Language. *IEEE Computer*, March 1985, pp. 11 - 19.

Booch, G. (1986). Object-Oriented Development. *IEEE Transactions on Software Engineering*, Vol. SE-12, No. 2, February 1986.

Conklin, J. (1987). A Survey of Hypertext. MCC Technical Report No. STP-356-86, Rev. 2, Software Technology Program, December, 1987.

Conklin, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer*, September 1987, pp. 17–41.

Cook, P., C. Ellis, M. Graf, G. Rein and T. Smith (1987). Project Nick: Meetings Augmentation and Analysis. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 132–146.

Cox, B. (1986). Object Oriented Programming – An Evolutionary Approach. Addison–Wesley Publishing Co., 1986.

Delisle N.M. and M.D. Schwartz (1987). Contexts – A Partitioning Concept for Hypertext. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 168–197.

De Marco, T. (1978). Structured Analysis and System Specification. Prentice-Hall Software Series.

Denning, P. (1982). Electronic Junk. *Communications of the ACM*, Vol. 23, No. 3, pp. 163-165.

DeSanctis, G. and R.B. Gallupe (1987). A Foundation for the Study of Group Decision Support Systems. *Management Science*, Vol. 33, No. 5, May 1987.

DIGITALK, Inc. (1986). SMALLTALK/V – Tutorial and Programming Handbook. Los Angeles, 1986.

Dobrowolski, G. and M. Zebrowski (1987). Ranking and Selection of Chemical Technologies: Application of SCDAS Concept. In: A. Lewandowski and A. Wierzbicki, Eds., *Theory, Software and Testing Examples for Decision Support Systems*, WP-87-26, *International Institute for Applied Systems Analysis*, Laxenburg, Austria.

Fikes, R.E. (1981). Odyssey: A Knowledge–Based Assistant. *Artificial Intelligence*, Vol. 16, pp. 331–361.

Fuhrmann, C. (1987). TELECTR User's Manual. IIASA Software Library Series, LS-16, *International Institute for Applied Systems Analysis*, Laxenburg, Austria.

Goldberg, A and D. Robson (1983). SMALLTALK-80: The Language and Its Implementation. Addison-Wesley Publishing Co.

Grefil, I. and S. Sarin (1987). Data Sharing in Group Work. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 187–211.

Hiltz, S.R and M. Turoff (1985). Structuring Computer–Mediated Communication Systems to Avoid Information Overload. *Communications of the ACM*, Vol. 28, No. 7, July 1985.

Huber, G.P. (1984). Issues in the Design of Group Decision Support Systems. *MIS Quarterly*, September 1984, pp.195-205.

Jacky, J. and I. Karlet (1986). An Object-Oriented Approach to a Large Scientific Application. *OOPSLA '86 Proceedings*, September 1986.

Jacky, J.P. and I.J. Karlet (1987). An Object-Oriented Programming Discipline for Standard Pascal. *Communications of the ACM*, Vol. 30, No. 9, September 1987.

Jarke, M., M.T. Jelassi and M.F. Shakun (1987). MEDIATOR: Toward a Negotiation Support System. *European Journal of Operational Research*, No. 3, September 1987.

Lewandowski, A. and A.P. Wierzbicki (1987). Interactive Decision Support Systems – The Case of Discrete Alternatives for Committee Decision Making. WP-87-38, *International Institute for Applied Systems Analysis*, Laxenburg, Austria.

Lewandowski, A. (1988a). SCDAS–Decision Support System for Group Decision Making: Short User's Manual. *International Institute for Applied Systems Analysis*, Laxenburg, Austria, to be published.

Lewandowski, A. (1988b). Distributed SCDAS – Decision Support System for Group Decision Making: Functional Specification. *International Institute for Applied Systems Analysis*, Laxenburg, Austria, to be published.

Malone, T.W., K.R. Grant, K.Y. Lai, R. Rao and D. Rosenblit (1987). Semistructured Messages Are Suprisingly Usefull for Computer–Supported Coordination. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 115–131.

Malone, T.W., K.R. Grant, F.A. Turbak, S.A. Brobst and M.D. Cohen (1987). Intelligent Information Sharing Systems. *Communications of the ACM*, Vol. 30, No. 5, May 1987.

Malone, T.W. (1987). Computer Support for Organizations: Towards an Organizational Science. In: *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J.M. Carrol, Ed., The MIT Press, 1987.

Marchionini G. and B. Schneiderman (1988). Finding Facts vs. Browsing Knowledge in Hypertext Systems. *IEEE Computer*, January 1988, pp. 70–79.

Pearson, M.L. and J.E. Kulp (1981). Creating and Adaptive Computerized Conferencing System on UNIX. In: R.P. Uhlig, Ed., Computer Message Systems, North-Holland, 1981.

Schaffert, C., T. Cooper, B. Bullis, M. Kilian and C. Wilpolt (1986). An Introduction to Trellis/Qwl. In *Object-Oriented Programming Systems, Languages and Applications 1986, Conference Proceedings*. Association for Computing Machinery, September, 1986.

Schneider, H.J. (1979). Formal Models and Practical Tools for Information System Design. Proceedings of the IFIP TC-8 Working Conference on *Formal Models and Practical Tools for Information Systems Design*, Oxford, U.K., April 1979. Published by North-Holland Publishing Co.

Schneiderman, B. (1987). Designing the User Interface: Strategies for Effective Human Computer Interaction. Addison–Wesley Publishing Co., 1987.

Shaw, M. (1981). ALPHARD: Form and Content. Springer Verlag.

Stefik, M., D.G. Bobrow, G. Foster, S. Lanning, and D. Tatar (1987). WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, April 1987, pp. 147–167.

Yankelovich, N., B.J. Haan, N.K. Meyrowitz and S.M. Drucker (1988). Intermedia: The Concept and the Construction of a Seamless Information Environment. *IEEE Computer*, January 1988, pp. 81–96.

Yonezawa, A and M. Tokoro (1987). Object-Oriented Concurrent Programming. The MIT Press.

A Implementation issues

In this part of the paper we will present the experimental implementation of SCDAS system with built–in document processing concepts presented in the first part of the paper. These purposes motivated the basic assumptions regarding the implementation technology:

- since design of *functions* of the system and their validation is the main purpose of building the prototype, with communication issues postponed for further studies, it was decided to build the prototype as *the single machine, single user system*, similarly like previous implementations of SCDAS (Lewandowski, 1988a),
- it was decided to build the prototype in highly modular way to simplify splitting various functional modules between the workstation and the server. To achieve such a high level of modularity it was decided to apply the object–oriented programming methodology and SMALLTALK as the implementation language.

We will not discuss in details problems of software design and implementation. There are such important problems like sharing of common objects in multicomputer environment, problems of concurrency control, design of multi-user data base and linking this data base with object-oriented environment as well as problems of internal data structures, organization of cooperation between software components, etc. These issues will be discussed in separate papers.

A.1 Data Browsers

Data browsers allow entering, changing, deleting and browsing information related to all basic components of the problem definition – i.e. list of attributes, alternatives and committee members. The data browser is organized as collection of columns (*subpanes*); each column is assigned for processing one component of selected data item. The user can freely choose the size of the browser, as well as its position on the screen. The data displayed in the browser can be scrolled using the pointing device (e.q. the mouse); the same pointing device can be used for selecting data items.

The browser consists of several *sensitive fields*. With each field there exists the associated menu activated with a pointing device. The content of this menu depends on the state of the SCDAS process, state of the data in a browser as well as the privileges of the user (conference president vs. ordinary participant). The following are the sensitive fields:

- The *TopPane* header containing the name of the browser and the name of the problem. The menu associated with the *TopPane* header (see Figure 1) allows manipulation of the pane – i.e. moving, resizing, closing, switching to other browsers etc.
- The *Field* headers contain description of the information displayed in the subpane. Menu associated with the subpane header (Figure 2) allows access to two kind of documents:
  - *Help* documents relating to the meaning of information contained in this particular subpane as well as help documents relating to methods for processing this information. According to the previously introduced therminology these documents belong to the category of *class help documents*.

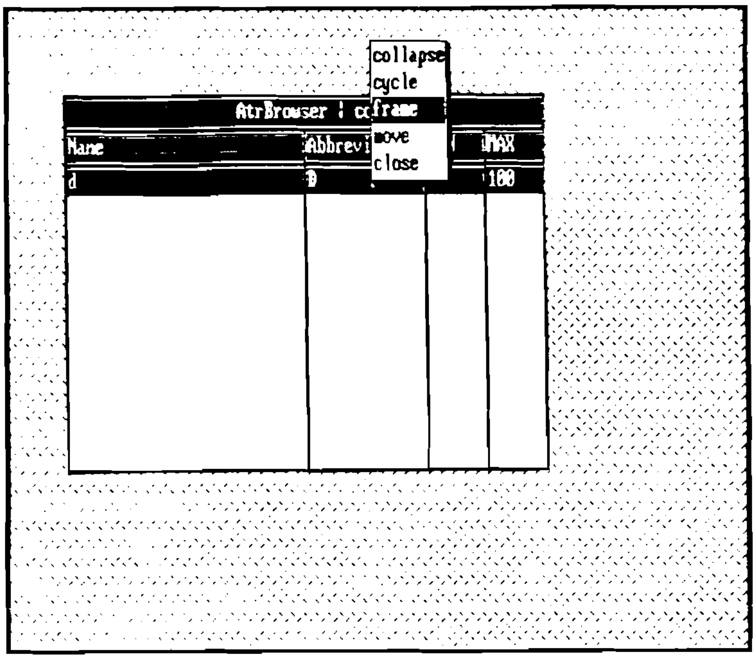

Figure 1: DataBrowser – TopPane menu

- Standard *Documents* containing information generated by committee president and by committee members and relating to the *general problems* concerned with definition, meaning, interpreting and processing data contained in a given subpane. It is necessary to mention, that these documents are also *class documents* and are relating to the *concepts* relating to a given data category, not the meaning of a given data instance. Therefore, in the subpane *Attribute* the committee president can distribute general comments addressing the issue, for example, *how we will discuss the attributes, how many attributes we will consider* etc.

- The *Data* field (subpane) containing data. The menu associated with subpane (Figure 3) allows data manipulation – adding new data items, deleting, searching etc. The exact form of this menu depends on the privileges of the user and state of the process – for example, the ordinary conference participant cannot add or delete new data to the browsers containing problem definition. Selecting items from this menu causes the appropriate action – like inserting or editing the content of selected data item in the subpane. One of the item in this menu is the *document* option. Contrary to header documents, these documents relate to *given instances of data.* These documents can contain comments generated by conference participants or by conference president and relating to a given, particular data item and they are *linked to data item currently selected as active (highlighted).* For example, in the case of attributes, documents relating to a given attribute can contain all information and comments regarding the exact meaning of this particular attribute; the alternative will have linked all documents and information describing properties of this given alternative, etc. Summarizing, documents accessible through the *Data field* belong to the category of *instance documents.*

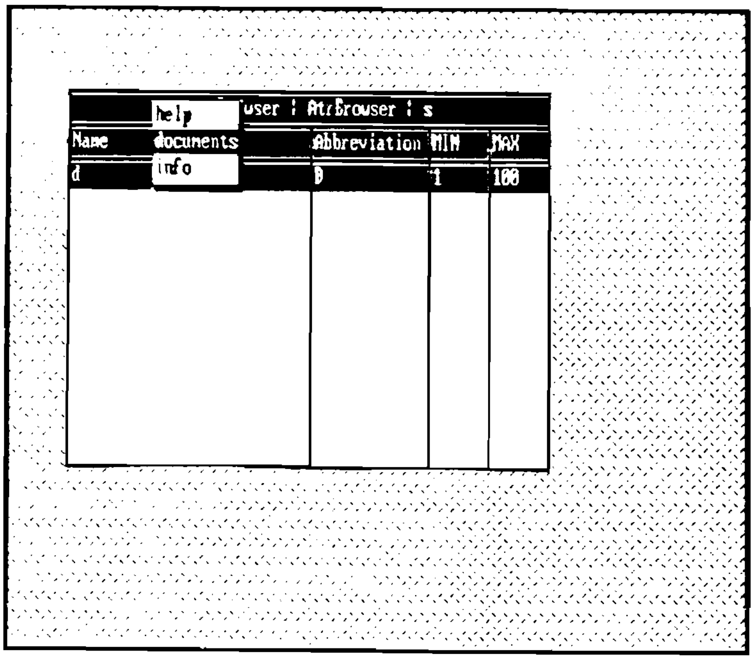

Figure 2: DataBrowser - subpane header menu

A.2 Document Browsers

The *Document Browser* gives access to all documents existing in the system except of help documents (see Figure 4). The user can select the *category* of document – like *Public, Messages, Note* and *Mail* category. All documents available in a given category are filtered according to privileges of the conference participant, status of the document and state of the process; list of these documents which are accessible to the user according to the current filtering procedure is displayed in the *Document* subpane.

The browser displays all information about document selected in the *Document* subpane – the *Link* subpane which tells about the data connected with the documents, the *Date* subpane which contains the information about creation date of the document, subpanes *Author* and *Title* containing information about the committee member who created the document and information when it took place as well as information about the *status* of the document. Finally, one of the subpanes is the *Text Pane* which is a standard text editor allowing access to the content of a selected document.

Similarly like data browser, the document browser has several sensitive fields which pointed by the mouse cause menu activation:

- The *TopPane* header which when pointed activates the similar menu, like in *Data Browser*. This menu allows resizing, moving and closing the browser window.
- The *TextPane* is the standard text editor which allows entering and modifying text, searching for a pattern, performing *cut* and *paste* actions as well as other functions typical for text processing. Special meaning has the *save* function which forces saving the document in internal data structure (the SMALLTALK image), and simultaneously performs all operations necessary to make the document accessible to other users, according to rules specific for the type of processed document. Normally, the area assigned

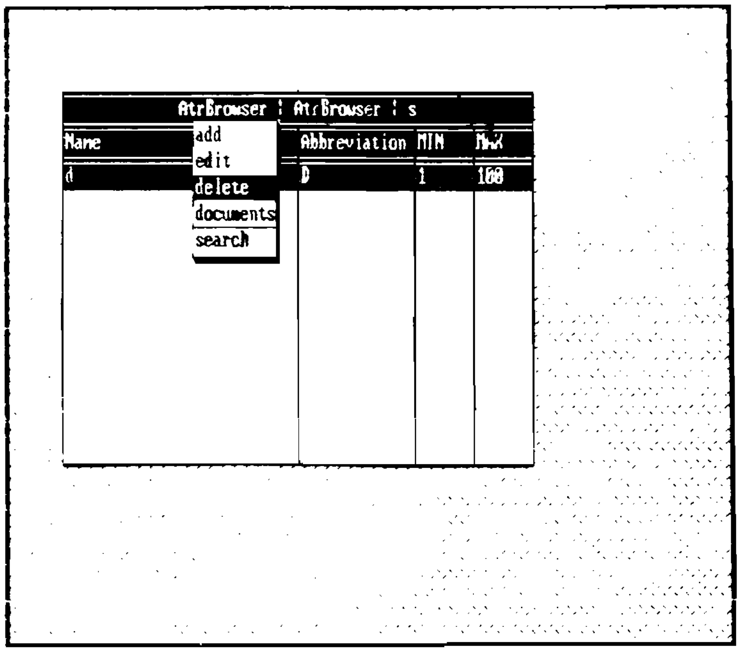

Figure 3: DataBrowser – data field subpane menu

for text editing is relatively small. It is possible, however, to perform the *zoom* operation which results in expanding this area to full screen size.

- The *DocumentTypePane* allows to select the category of documents being displayed in text pane. The menu associated with this pane depends on type of document and previous actions undertaken by the user:
  - Every document available to the conference participant can be copied and the copy moved to the *Note* category. This option allows to collect important information and protect this information against changes which could be done by the conference president or other users (e.q. deleting *public* or *message* documents).
  - It is possible to *mark* the selected document as the *node*. This action is necessary to create *links* between documents. When a certain document is marked, it is possible to select the other one. In this case the menu associated with *DocumentTypePane* is changed - instead of *marking document* it allows to *create a link*. When link is created, the marked document is automatically unmarked.
  - When the list of documents relevant to a selected one (i.e. the *node* associated with the selected document) is not empty the option *Browse Relevant Documents* is available in the menu. Selection of this option opens the *NodeBrowser* which allows to inspect documents being members of a given node.
- The *DocumentPane* contains the list of all documents belonging to the category selected in the *DocumentTypePane* and accessible to the conference participant. Selecting one item from the document list causes the update of all associated panes – i.e. the *TextPane* and all information panes containing information about the author, creation date, etc.

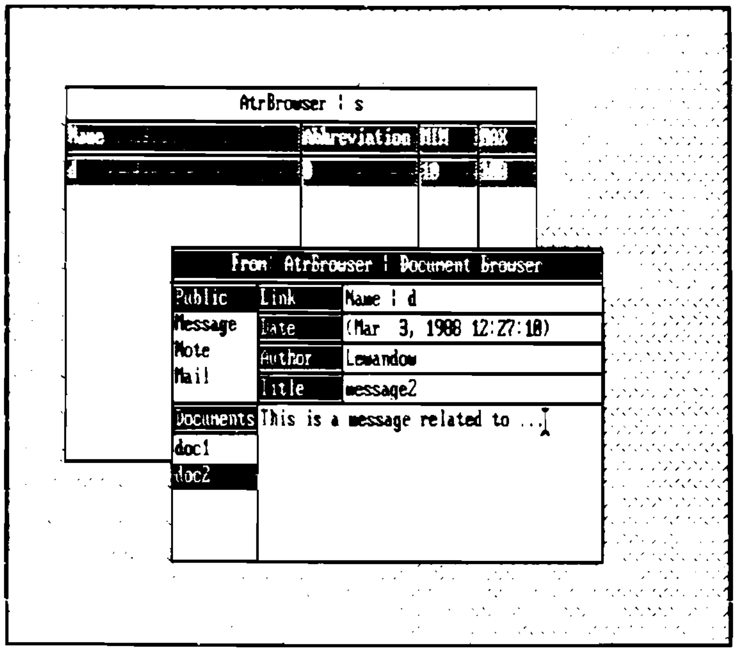

Figure 4: Document Browser

The menu associated with this pane, except of *add* and *delete* items allowing entering and deleting a document contains one additional item:

- The *Send* option which is available only for *Mail* documents created by the conference participant. Selecting this option triggers creation of copy of the document and sending this copy to the receiver. Such a document, when selected will display its status as *Sent.*
- The *Broadcast* option which is available only for *Message* documents created by the conference participant. Selecting this option changes the status of the document; such a document will be available to all participants (including the author) in *read-only* mode. Such a document, when selected will display its status as *Broadcasted.* The only person who can modify and delete broadcasted document is the conference president.

A.3 Help Browser

The Help Browser allows displaying and modifying the help information (Figure 5). This browser is a simpler version of the *DocumentBrowser* – it consists only of the *DocumentPane* and the *TextPane.* Due to relative simplicity of the SCDAS system it was decided, that help documents will be not organized in a hypertext–like hierarchy. These documents constitute the simple list of documents linked to a given section of the browser. All users can contribute to the pool of help documents and can freely modify documents which they have generated. Help documents can be, however, used as elements of the linking procedure.

It is necessary to point out, that the user has the full freedom in organizing the information presented on the screen. The number of browsers existing the same time is unlimited;

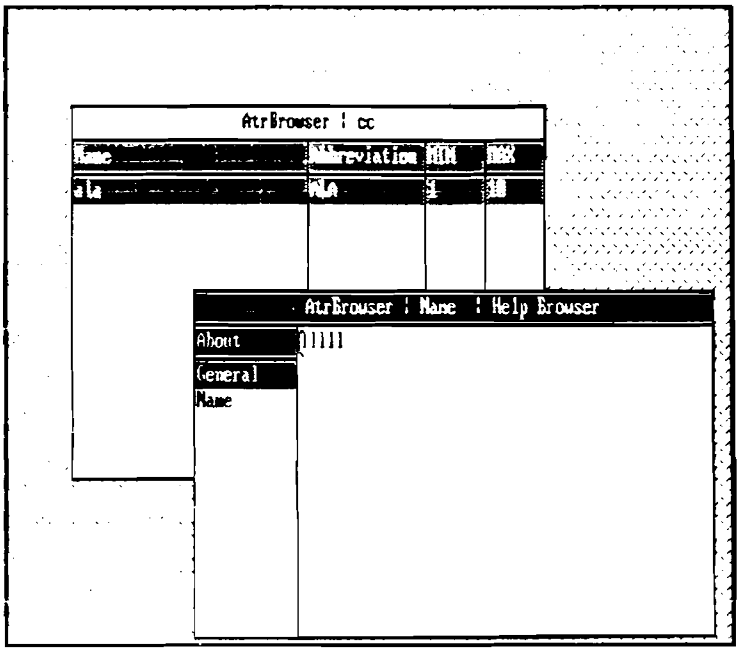

Figure 5: Help Browser

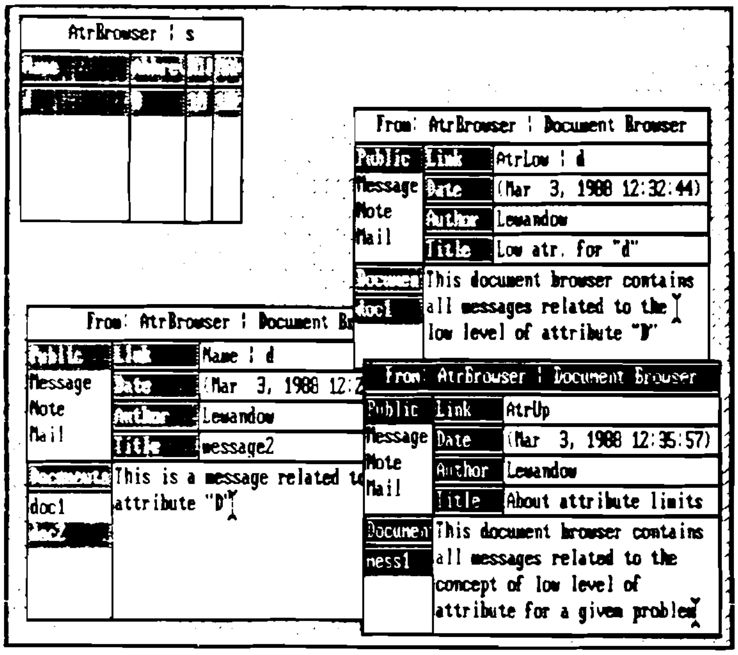

Figure 6: Multi-window information presentation in SCDAS

their shape and location can be freely selected by the user (Figure 6). This flexibility of presentation has been achieved due to the features offered by the powerful Smalltalk interactive environment. Such a flexibility can result in overloading the user – it is very easy to be lost if computer screen contains too much information. Experience with other windowing environments shows, however, that after certain period of learning the user is able to find the best way of information presentation for a given decision problem and for his own information processing capabilities.