Organizational Memories as Electronic Discussion By-products

Natalia A. Romero, José A. Pino, Luis A. Guerrero
Departamento de Ciencias de la Computación
Universidad de Chile
Casilla 2777, Santiago, Chile
{nromero, jpino, luguerre}@dcc.uchile.cl

Abstract

An approach to develop Organizational Memory Systems is presented. It is based on the re-use of computer-supported discussions held as meeting preparations. The discussions are rich in informal knowledge, which is basic for a good Organizational Memory. The presentation is illustrated with a brief introduction to OMUSISCO, a software system based on the approach. OMUSISCO uses SISCO meeting preparation discussions as information source. A thesaurus is built to help users access knowledge. The system was tested with simple discussions but it should scale up well in real situations.

Keywords: Organizational Memory Systems, Pre-meetings, SISCO, Thesaurus.

1. Introduction

Organizational Memories (OM) are mechanisms to preserve, distribute and reuse the knowledge accrued by an organization. Conklin [9] has emphasized the competitive advantages for the organization both in the short and long term when this memory is available.

This paper presents our approach to develop an Organizational Memory System (OMS), which is based on the automatic information capture from computer-supported discussions. The information is then structured to become knowledge which can be easily accessed.

Schwartz et al. [16] have identified three main functionalities an OMS must have:

- *Acquire:* gather – inquire – validate/verify – encode. It is concerned with how to collect knowledge from people and other resources, and store it in an Organizational Memory.
- *Organize:* profile – associate – rank – classify. It refers to structuring, indexing and formatting the knowledge to be able to search and find it.
- *Distribute*: it is the ability to provide the relevant information to the person who needs it at the right time.

The most difficult functionality to provide is certainly the first. Approaches based on voluntary submissions of knowledge to the OM are probably too optimistic: people do not have the time nor the motivation to spend a significant amount of time to provide information just for the eventual use of unknown people in the future. Another approach with little chance of success is based on knowledge engineers working with people to extract knowledge: besides the high cost such approach might have, the product might be unacceptably biased according to the engineers' own model of what is important to remember. Finally, simple procedures to store formal documents such as letters and invoices are doomed to failure in capturing interesting knowledge which might be in people's minds rather than on paper.

One challenge is then to gather valuable knowledge from people with the least possible overhead on them. Such knowledge should include tentative projects, arguments in favor or against the development of such projects, concepts about the organization itself, its crises and how they were overcome, experience with the products or services provided, etc. One possible source for such information is electronic mail messages. However, information in e-mail messages does not have structure.

An opportunity to get the knowledge we are seeking is perhaps provided by stored electronic discussions and meetings within an organization. In particular, Borges et al. [6] have developed a system called SISCO to support asynchronous distributed discussions. The supported discussions are intended as preparations for face-to-face decision meetings to occur afterwards. The rationale for having one such discussion is that if it is held over some period, all issues can be appropriately debated, most

arguments and counter-arguments will be given, supporting data will be provided, and people will have the necessary time to mature ideas and ponder options. Furthermore, as known from previous Electronic Meetings research [13], participants will have more freedom to express themselves, especially if anonymous inputs are allowed. Finally, the asynchronous distributed way of working lets participants contribute to the discussion from anywhere at anytime, in line with decision makers' mobility and busy schedules.

SISCO has another valuable feature for our OMS: the supported discussions are structured. The structure is useful for the discussion, of course, but for our purposes it allows to capture the context and the type of a contribution, simplifying the task of structuring information for the OMS. SISCO uses an enlarged IBIS [12] discussion model.

In what follows, we describe OMUSISCO, an OMS which is fed with SISCO discussions in an automatic way and provides access to an OM built on them. The main goal to develop OMUSISCO was to have an OMS based in a specific information source. A set of algorithms to automatically generate and maintain the structured information had to be developed. Such system may have the following applications:

- *Provide early learning support.* New employees can benefit from accessing knowledge accrued by the organization.
- *Provide memories of past experiences.* People may review previous decisions and the context in which they were made, successful and failed projects, etc.
- *Share individual knowledge.* People may access ideas, data, arguments, etc. provided by employees in the past.

2. Organizational Memory

The OM concept has evolved with time. The first model of an OM considers a set of documents and information pieces from the organization's history stored in a common space (repository) that can be brought to bear on present decisions [20]. A second model enlarges the previous one by including information generating processes, since these provide the context within which the information was originated. Thus, experiences, conversations, decisions, etc. are the background for the information artifacts [8]. Stein and Zwass emphasize impact of the previous knowledge on present activities, resulting in higher or lower levels of organizational effectiveness [19]. Jennex [11] views OM as a combination of concrete and abstract knowledge: the first is the history and trend data collected and the second class of knowledge is the experience gained by the organizational member over time.

Recent approaches view OM as a repository of information artifacts + processes + individual memories coming from human agents or other systems. Thus memory is a "limited object" which is an entity but at the same time it may be embedded in many organizations and individual processes where it may have several meanings depending on the host [2].

An important issue when developing an OMS is to ask the requirements to be fulfilled by a successful OM. Among the most relevant ones are to reach the appropriate information sources and to have a cooperative social environment in the organization [1]. We have already mentioned that the first requirement will be satisfied with the use of SISCO discussions. With respect to the second one, it can only be stressed it does not make sense to build an OM if there is a bad social environment in the organization: people will simply not contribute during meeting preparation because they probably distrust the use of those contributions afterwards. The extreme case is what we call "the Nixon's syndrome": people being afraid anything they have said may be used against them (in reference to the famous voluntarily recorded tapes from conversations by former US President R. Nixon himself during the Watergate trial). It is an open issue to know the number of organizations in which this assumption holds.

3. Gathering information

The simplest approach to capture information one may try is to process plain text available in many sources in the organization: letters, reports, electronic mail messages, memoranda, etc. This seemingly simple strategy has an important problem: the information has poor context, and it is very difficult to associate a meaning to many words, phrases, sentences or paragraphs when there is no context. Much tacit knowledge was probably added to such information pieces when originally decoded in order to make it meaningful but such tacit knowledge is not automatically available later on. Therefore, what appeared to be a simple approach can get complicated if the goal is to develop an OMS with greater added value than a simple text retrieval system with keyword matching.

One way of circumvent the absence of tacit knowledge is to try to obtain contextual clues from information originally structured by the authors. For example, let us consider the IBIS model of discourse [12]. In IBIS, all information is of one of three types: issue, position and argument. Issues are questions, positions are answers to such questions, and arguments are chains of reasoning supporting or objecting positions. Issues may induce new questions. The conversation (discussion) then is a sequence of questions with their associated possible

answers and arguments. Of course, the conversation does not need to proceed ending an issue before going to the next one; on the contrary, participants may add information of any type to any issue at any time. There is no deletion of information and thus the discussion gets more comprehensive as time passes.

It is easy to extract context from an IBIS discussion, since at least the type of information is clearly known and also the related information (e.g., current and neighboring issues). Of course, this help is not free: the discussion participants have paid the cost as additional work trying to organize their contributions in the required types and keeping all discussion in the required framework.

SISCO uses an enlarged IBIS model as its information structure [5], since it needs additional information types in its decision making preparation discussions. Besides issues, positions and arguments, there are pre-decisions, proposals, tasks, remarks and infobase. Pre-decisions are assumptions or common agreements made beforehand; a proposal is an issue specialization used to suggest a task to be carried out; a task is the description of an activity to gather some additional data needed for the rest of the discussion; this extra data is stored in the infobase; finally, the remarks are statements which do not clearly fit in the other elements. Furthermore, a SISCO discussion is organized as a hierarchy. The chapters of the discussion are the *agenda items.* Each item has a set of *objectives,* each of which has a set of discussion elements (issue, proposals, etc.). Figure 1 shows this hierarchy.

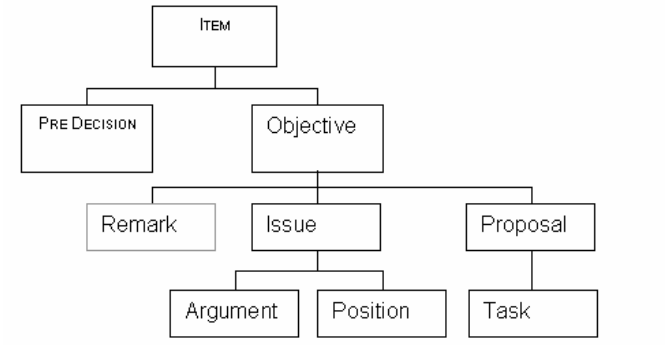

Figure 1. Document: minimal information unit

4. OM Structuring

The rich SISCO structuring is an asset that should be exploited, but it is not the structuring one needs to use in the context of an OM. One design decision which has to be made is whether to build a completely new structure extracting knowledge from the SISCO databases or not.

Our choice was to keep the original SISCO structures and to enhance them to serve the needs of the OM. In this way, the SISCO databases are held intact, but the information is accessed through other means using OMUSISCO. Figure 2 illustrates this approach.

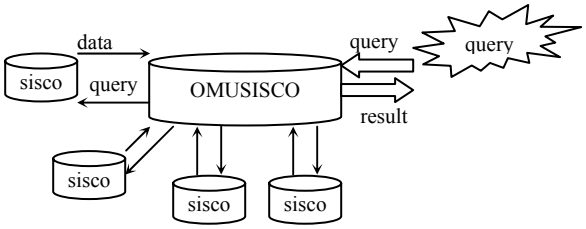

Figure 2. Queries to OMUSISCO information

Users have full real-time access to the SISCO discussions and thus, what is needed is to add an associative memory capturing the intended semantics of terms in the SISCO databases. Therefore, essentially three new entities were added:

- *Term:* vocubulary components. We need to store occurrence frequencies.
- *Document:* part of the discussion we would like to identify as a unit. It is the hierarchy shown on Figure 1. We need to store the relation of this document with the captured terms.
- *Element:* discussion element we would like to associate to terms.

Figure 3 shows the additional structure we designed for the OM derived from SISCO discussions with these entities. It incorporates terms (a concept which was not considered in the SISCO structures) and their relationships to other structural components. The structure is automatically fed with information from the SISCO databases, automatically updating the OM.

5. OM Access Structures

The goal now is to develop structures to access information stored in the SISCO discussions based on keywords and phrases provided by the user. The additional structure described in the previous section should be very useful for this task. In fact, TERMDOC and TERMELEMENT (Figure 3) are implementations of the *inverted file* concept. In standard Information Retrieval, the inverted file is a mechanism for indexing a text collection in order to speed up the searching task [3].

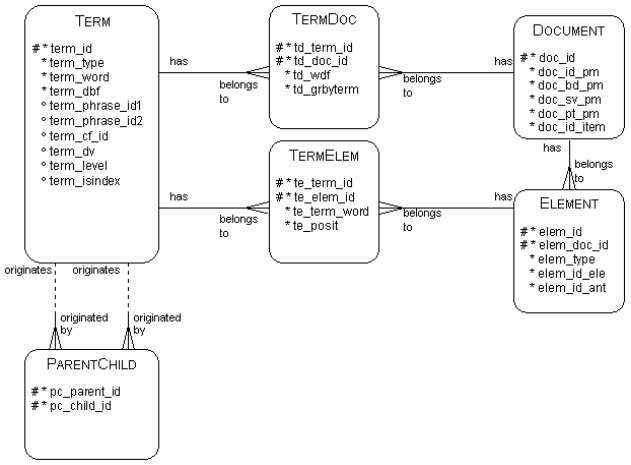

Figure 3. OMUSISCO knowledge structure

In Information Retrieval, the access unit is a "document". In our case, "a document" must be defined as an atomic piece of information from a SISCO database with enough context data to be useful. Analyzing the SISCO structural elements and their relationships, the natural choice for the definition of a document is the structure shown in Figure 1, since it represnts a conceptual unit.

As it is well-known in Information Retrieval, a thesaurus (i.e., a terms list with information about their use and the relationships among them [14]) is very useful when building an associative memory. Its main applications are [10]:

- to provide a standard vocabulary for indexing and searching,
- to assist users locate terms for appropriate query specification, and
- to provide a hierarchical classification to allow enlargement or narrowing of a query according to the user needs.

A literature search was done to find the best algorithms to build and maintain the thesaurus. An interesting related work is the research done by Chen et al. [23] to identify the important topics of electronic brainstorming sessions of GroupSystems [13]. They used automatic indexing, cluster analysis and Hopfield net classification.

We decided to use *index terms* to locate the useful information. These terms are stored in the thesaurus together with the rest of the vocabulary with the relationships among them. For the automatic construction of the thesaurus, we proceeded as follows:

The first step was to prepare the vocabulary [4]. The lexical analysis selects words from the text. Then, the words are filtered by deleting stopwords (frequent words: prepositions, adverbs, etc.). Finally, there is a *stemming* process, in which stems or parts of words are derived form words to improve retrieval efficiency and decrease index size.

The second step was to assign term values and according to them choose the terms to be included as indexes in the thesaurus. From the variety of available methods, we chose two:

- selection by occurrence frequency [14]. Each term is placed in one of three categories according to its frequency: high, medium and low. Medium frequency terms are the best for indexing and search. Two frequency thresholds are needed as input from the user: low to medium and medium to high.
- selection by discrimination value (DV) [15]. DV measures the degree with which a term is able to distinguish among documents in the collection. The higher is the DV for a term, the better is the term as an index. It is necessary to compute the average similarity for the documents of the collection with the term and then without the term being evaluated. Then, for a term k:

$$DV(k) = (Average\_similarity\_without\_k) - (Average\_similarity\_with\_k)$$

The third step was to build phrases. It is an optional step, used whenever the user wants to have pre-coordinated terms [14]. Some phrases can get medium frequency when built around high frequency terms, and thus, this is a way to increase the number of useful terms.

The algorithm generates phrases form pairs of terms with distance 0 using calculations of co-occurrence and cohesion. Distance 0 is defined as two terms in the same element without other relevant terms between them.

The fourth step was the similarity computation, following Srinivasan's procedure [18]. This step computes the statistical similarity between pairs of terms. Two methods were incorporated:

- *Cosine.* If p and q are the number of documents associated to two terms, this measure is computed as:

$$Cosine(p,q) = \frac{coocurrence(p,q)}{\sqrt{p * q}}$$

- *Dice.* If p and q are defined as above:

$$Dice(p,q) = \frac{coocurrence(p,q)}{p+q}$$

The fifth and final step is the organization of the vocabulary. Following again Srinivasan's procedure, we did not include the usual clustering process. Instead, the simple Forsyth and Rada's procedure is used [17]. The algorithm is based on building hierarchical relationships among terms. Two assumptions are made: i) high-frequency words have broad meaning, while low-frequency words have narrow meaning, and ii) if the density functions of two terms p and q (of varying frequencies) have the same shape, then the two words have similar meaning. With these two assumptions, if p is the term with the higher frequency, then q becomes a child of p.

6. OMUSISCO: a tool to support OM

The architecture of our system is shown in Figure 4: a query stated by a user is enhanced with the help of the thesaurus in order to use index terms; these are input to the inverted index to locate relevant documents; after retrieving documents from the SISCO databases, they are presented to the user who may modify the query to start a new cycle of the retrieving process. The user may browse related documents according to the SISCO structures. Due to the definition of a document, a user typically will retrieve all the relevant contextual information. However, if he needs additional information following the SISCO structure (e.g., if he gets interested in the whole discussion in which the document is inserted), he may connect to SISCO and access the target with the data provided by OMUSISCO.

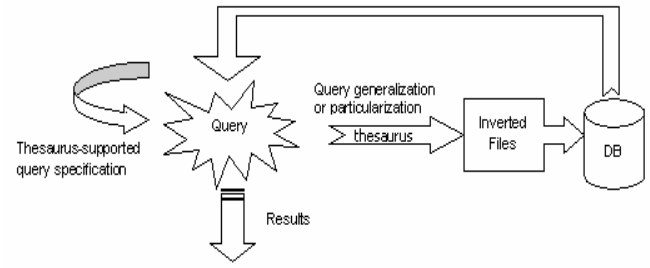

Figure 4. OMUSISCO services

OMUSISCO is a system implementing this approach. The software has two modules: one oriented to knowledge management and the other to provide end-user information retrieval.

6.1. Management Module

The objective of this subsystem is to maintain the OM, specifically the thesaurus and the consolidated database. It must allow insertion of new pre-meeting discussions. It also has to be able to update indexes to reflect the merging of knowledge from new discussions with previous one.

One interesting design parameter is the degree of involvement we require from the person responsible for this function. On one extreme, the system may be fully automatic, but this implies the administrator can not incorporate his knowledge of the organization. The other extreme is a system where the administrator has full control over the contents of the OM, but he may have excessive work to do.

Our design lets the administrator know the list of terms and phrases describing the OM. He may customize it according to his own experience with the organization's knowledge. His work improving the indexes has direct effect on later access and retrieval performance.

The current OMUSISCO implementation allows customization by the administrator only by deleting terms and phrases. A second version can include features to add new terms and phrases. Figure 5 shows a menu to automatically generate a thesaurus using the procedures described in Section 5 above.

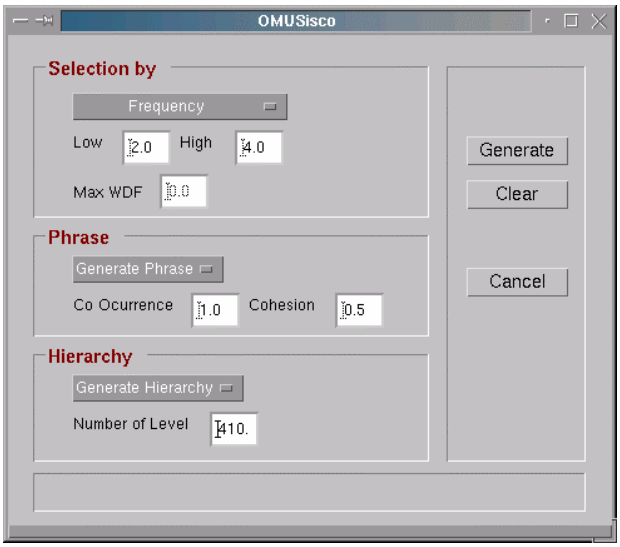

Figure 5. Management module menu

6.2. Information Retrieval Module

This module should allow users to get access to the knowledge they are seeking. Traditional information retrieval tools let users specify search terms and the system provides a list of *hits:* documents satisfying the search criterion. Our design follows this tradition (Figure 6).

In our first version, queries consist of simple terms (Boolean expressions of these can easily be added in a second version). Use of the thesaurus allows the search to be done with index terms. The query may also be enriched

or narrowed: the thesaurus suggests further index terms within context the user may accept or not. For instance, the query may be required to be performed only on *issues*.

The user may customize the system in two ways. First, the user may temporally disable terms from the OM which may be considered useless or disturbing for the queries he is stating. Second, the system lets the user choose search algorithms and parameters.

The results are shown as a list of hits. By choosing one of them and depressing the "Detail" button, one gets access to the corresponding SISCO element; the system also gives details about the neighboring SISCO elements providing a contextual hierarchy of the retrieved elements.

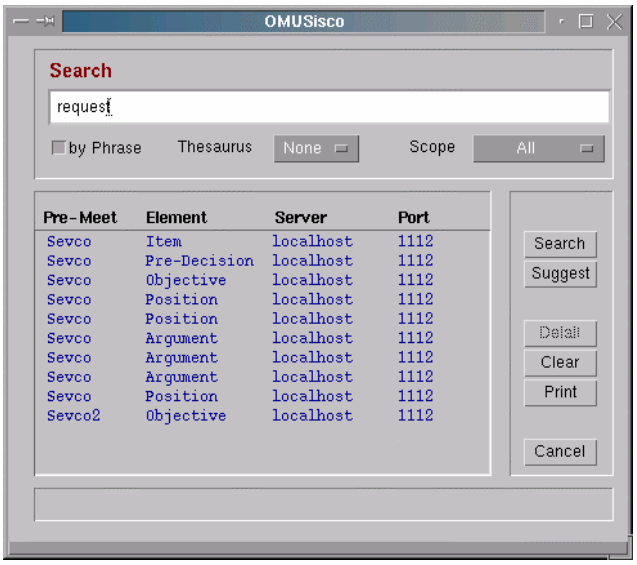

Figure 6. Query specification and results

6.3. System Construction

OMUSISCO is a client/server application. The client was programmed in Java 1.18 (using applets) and the database server is MiniSql 1.0. The system runs under linux.

The heart of the system is the automatically generated thesaurus, which is used both for indexes generation based on contents and for supporting user queries. The thesaurus was implemented with algorithms available from the literature, which were adapted to work in the OMUSISCO environment.

7. Results and Discussion

The first version of the system was tested with three SISCO discussions. As a SISCO implementation we used USISCO 1.0 [21]. Two of the discussions were extracted from the SEVCO Industries case described in [17] and the third was an example discussion made by ourselves.

The results are promising based on these tests, since the efficiency is high. Both in terms of disk space used by OMUSISCO and time used to access the information, the system is very efficient. It is interesting now to consider performance in real case situations. We do not have experiments yet, but a discussion can be held beforehand.

The first issue is disk space used by OMUSISCO in addition to the SISCO discussions themselves. It would not be desirable this space would grow very fast compared to the space used by the discussions. Fortunately, the additional space is mainly to store the thesaurus and this stores a subset of all different words present in the SISCO databases (plus pointers). Now, the Heaps' law [22] states that the vocabulary of a text of size n words is of size $V = K\, n^{\beta}$, where constants K and β depend on the particular text. K is usually between 10 and 100 and β has common values between 0.4 and 0.6. Therefore, the vocabulary grows in a proportion close to the square root of the text size. As a consequence, we do not have to worry much about the disk space demands posed by OMUSISCO in real cases with large discussion databases.

The second issue concerns time efficiency during retrieval. It is well known inverted indexes are very fast on search. Searching a simple word in 250 Mb of text using an inverted index took 0.08 seconds on a Sun UltraSparc-1 of 167 MHz [3]; searching a phrase in the same setup took 0.25 to 0.35 seconds. In OMUSISCO, the inverted index is small, since it stores only few words of the vocabulary (the index terms). Besides, the number of acceses to the database is minimal. Thus, again, we should be reasonably confident OMUSISCO should perform well in a real world case.

8. Conclusions

We have presented an approach to Organizational Memory based on pre-meeting discussions and information retrieval. The design has been implemented in OMUSISCO with which we have made some very simple tests. Although we have not made real-life tests, we can anticipate the performance of the system will not be bad.

Another type of evaluation, of course, has not been done. It concerns the assumptions we have made. Will people confidently submit SISCO contributions knowing these texts may be used afterwards? (a worst case is what we called Nixon's syndrome: "everything you have said can be used against you"). Will people accessing an OM be ready to state their needs in terms of search words and phrases? Do people really discuss interesting subjects in computer-supported meeting preparation?

These questions generate other important issues. Is it ethical to store contributions intended for a discussion eventually for ever? Is it right to allow access to a discussion database to other people than the ones originally participating in the discussion? These are very crucial issues which must be answered before proceeding to actual use of the system. At the very least, users should be informed when participating in discussions about the later uses of their contributions. Anonymous contributions may also be suggested as a way to avoid later misuses of the information and to encourage users to contribute at the same time.

It is interesting to note these ethical issues are not only applicable to SISCO-OMUSISCO. In general, any stored comments can be potentially searched and retrieved afterwards. Nevertheless, the technical literature on computer-supported discussions and brainstorming has not mentioned this caveat: see, e.g., [13, 23, 24].

In a study done to determine the requirements for a Knowledge Management system, Carstensen and Snis [7] state nine requirements: recording, communication, refinement, annotation, classification, retrieval, navigation, context-sensitivity and multi-modality. All of these (except the last one) can be provided by SISCO/OMUSISCO.

Concerning the multi-modality, which is required is the possibility of incorporating knowledge in various media and both formal and informal knowledge (Conklin had also asked for formal and informal knowledge for an OM [9]). We have not worked on multimedia knowledge yet, although theoretically there are no reasons why this could not be incorporated into OMUSISCO (in fact, there is much research being done on multimedia retrieval).

Formal knowledge - as contained in letters, invoices, memoranda, etc. - could also be included in a future OMUSISCO version. A simple way to achieve this is to extract terms and phrases from the documents, build the access structures as with discussions, and allow access to the documents when a user requests them after seeing results from a query.

Acknowledgements

We thank suggestions to improve the paper provided by three anonymous referees. This work was partially supported by grant No. 1000870 from FONDECYT (Chile).

References

[1] Abecker, A., Bernardi, A., Hinkelman, K., Kuhn, O., Sintek, M.: "Towards a well-founded technology for Organizational Memories". German Research Center for Artificial Intelligence – DKFI GmbH, www.dfki.de, 1998.

[2] Ackerman, M., Halverson, C.: "Reexamining Organizational Memory". *Communications of the ACM* 43, 1 (2000), 59-64.

[3] Baeza-Yates, R., Ribeiro-Neto, B., Navarro, G.: "Indexing and Searching". In R. Baeza-Yates, B. Ribeiro-Neto (eds.): Modern Information Retrieval, Addison Wesley – ACM Press, Harlow, England, 1999, 191-228.

[4] Baeza-Yates, R., Ribeiro-Neto, B., Ziviani, N.: "Text Operations". In R. Baeza-Yates, B. Ribeiro-Neto (eds.): Modern Information Retrieval, Addison Wesley – ACM Press, Harlow, England, 1999, 163-190.

[5] Bellassai, G., Borges, M., Fuller, D.A., Pino, J.A., Salgado, A.C.: "An IBIS-based model to support group discussions". In B. Glasson, D. Vogel, P. Bots, J.F. Nunamaker (eds.): Information Systems and Technology in the International Office of the Future, Chapman and Hall, London, 1996, 49-62.

[6] Borges, M.R., Pino, J.A., Fuller, D.A., Salgado, A.C.: "Key issues in the design of an asynchronous system to support meeting preparation". *Decision Support Systems* 27 (1999), 269-287.

[7] Carstensen, P.H., Snis, U.: "On Knowledge Management: A Field Study". In D.G. Schwartz, M. Divitini, T. Brasethvik (eds.): Internet-based Organizational Memory and Knowledge Management, Idea Group Publishing, Hershey, PA, USA, 2000, 170-199.

[8] Conklin, E.J.: "Capturing Organizational Memory". In D. Coleman (ed.): Groupware'92, Morgan-Kaufmann Pubs., San Mateo, CA., 1992, 133-137.

[9] Conklin, E.J.: "Designing Organizational Memory: Preserving intellectual assets in a Knowledge Economy". http://www.gdss.com/DOM.htm.

[10] Foskett, D.J.: "Thesaurus". In K. Sparck Jones and P. Willet (eds.): Readings in Information Retrieval, Ellis Horwood, West Sussex, England, 1986.

[11] Jennex, M.E.: "Using an Intranet to manage knowledge for a virtual project team". In D.G. Schwartz, M. Divitini, T. Brasethvik (eds.): Internet-based Organizational Memory and Knowledge Management, Idea Group Publishing, Hershey, PA, USA, 2000, 241-259.

[12] Kunz, W., Rittel, H.: "Issues as elements of Information Systems". Working Paper #131, Institute of Urban and Regional Development, U. of California at Berkeley, 1970.

[13] Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., George, J.F.: "Electronic meeting systems to support group work". *Communications of the ACM* 34 (1991), 40-61.

[14] Salton, G., McGill, M.: Introduction to Modern Information Retrieval, McGraw Hill, New York, NY, 1983.

[15] Salton, G., Yang, C.S.: "On the specification of term values in automatic indexing". *Journal of Documentation* 29,4 (1973), 351-372.

[16] Schwartz, D.G., Divitini, M., Brasethvik, T.: "On Knowledge Management in the Internet age". In D.G. Schwartz, M. Divitini, T. Brasethvik (eds.): Internet-based Organizational Memory and Knowledge Management, Idea Group Publishing, Hershey, PA, USA, 2000, 1-23.

[17] Senn, J.A.: Analysis and design of information systems. McGraw-Hill, New York, 1991.

[18] Srinivasan, P.: "Thesaurus construction". In W. Frakes, R. Baeza-Yates (eds.): Information Retrieval – Data structures and Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 1992, 161-218.

[19] Stein, E.W., Zwass, V.: "Actualizing Organizational Memory with Information Systems". *Information Systems Research* 6,2 (1995), 85-117.

[20] Walsh, J.P, Ungson, G.R.: "Oganizational Memory".*Academy of Management Review* 16,1 (1991), 57-59.

[21] Espinosa, J., Pino, J.A., Pollard, P.: "On the development of a SISCO implementation using Java". Proc. of CRIWG'97, 3rd. International Workshop on Groupware, El Escorial, Spain, Oct. 1997, 17-24.

[22] Heaps, J.: Information Retrieval – Computational and Theoretical Aspects. Academic Press, 1978.

[23] Chen, H., Hsu, P., Orwig, R., Hoopes, I. and Nunamaker, J.F.: "Automatic Concept Classification of Text". *Communications of the ACM,* 37(10), 1994, 56-73.

[24] Shirani, A., Aiken, M. and Paolillo, J.: "Group decision support systems and incentive structures". Information & Management 33, 1998, 231-240.