Structural Computing in the Collaborative Work Domain?

Jörg M. Haake

GMD - German National Research Center for Information Technology
IPSI - Publication and Information Systems Institute
Dolivostr. 15, D-64293 Darmstadt, Germany
haake@darmstadt.gmd.de

Abstract. Structural computing is a new paradigm for developing applications in new domains. One of its benefits is that adaptation of behavior—as a consequence of changes of the structures used to model the application—becomes easier. In this paper, the collaborative work domain—in particular the task of coordination—is examined as a potential application area for structural computing. Coordination behavior shows high variability, depending on the ever-changing requirements of cooperative work processes. This makes frequent adaptation of the coordination behavior necessary. To support this adaptation, a structural computing approach is described in this paper, which (1) explicitly models processes, teams, and content as part of a shared workspace, and (2) dynamically defines coordination support as behavior (i.e. computations over the above structure). Finally, some requirements on structural computing, which result from this work, are presented.

1 Introduction

As Ken Anderson points out in his introduction to the 2nd International Workshop on Structural Computing (San Antonio, Texas, June 3rd, 2000), structural computing "asserts the primacy of structure over data and aims to develop a set of principles, techniques, and technology that can be used by developers as resources for developing applications in new domains" (as introduced by Nrnberg et al, [16]). In this position paper the domain of collaborative work is examined as a potential application area for structural computing.

In the past, research on collaborative work has focused either on understanding how people work together (e.g. using computers as supporting tools or as a medium) or on how to develop computer-based tools for teams (e.g. groupware). Most researchers looked at specific tasks or domains, where people need to work together. As a result, some tasks can now be supported by specific groupware tools or environments. Other researchers focused on identifying general tools, which address basic tasks to be performed in any collaboration (such as communication, coordination and manipulation of shared artifacts). As a consequence of the generic approach, these generic tools usually need to be adapted to the domain or task at hand. Otherwise, they usually lack a specific ease of use (e.g. due to missing domain specific support).

Adaptation of generic tools to a specific task is not easy. This problem becomes even more critical when properties of the task are frequently changing, and these properties changes should be reflected in new adaptations. In this case, adaptation becomes a moving target.

One such task is coordination in shared workspaces, where teams create and manipulate shared artifacts to solve a complex problem (e.g. to develop software, to design a product, or to write a research paper). As described above, dedicated collaboration environments (e.g. collaborative software development environments, or shared document editors) have traditionally supported such tasks. However, the task of coordination is still a problem.

One possibility of supporting coordination in such a shared workspace arises when looking at the coordination problem from the structural computing point of view. The shared workspace may capture shared artifacts describing (partial) solutions as well as plans and arguments of how to achieve the final result. Events raised by actions of co-workers change the state of the shared workspace. Coordination can then be described as the task of minimizing differences between the actual state and the terminal state to be achieved over a number of steps (occurring over the lifetime of the collaboration). Letting the structure of the workspace and the structure of the events being executed on that workspace drive the coordination support might actually devise a new approach to coordination of shared work.

In this paper, first the properties of the coordination task in shared workspaces are analyzed. Then, previous approaches to coordination are summarized. Next, some possibilities of supporting the coordination task with a structural computing approach are examined. Finally, some conclusions for future work are presented.

2 The Coordination Task in Shared Workspaces

Haake [9] defined a "shared global workspace" as an arbitrary hypermedia structure, which is used by multiple teams or users to facilitate asynchronous and synchronous collaboration. Not all hypermedia objects in such a shared global workspace are in use all the time. Generally, the users may use parts of the shared global workspace as shared local sub workspaces. A shared local workspace can be used for supporting either synchronous collaboration (i.e. a shared workspace offering synchronous sharing of objects and views) or asynchronous collaboration (i.e. only one user works on the objects in the local workspace and other users will access the results only later).

Work in a shared global workspace can be viewed as a sequence of state transitions over time. Each state describes the current properties of the shared global workspace such as the individual states of nodes and links (e.g. information objects, documents and their relationships) contained in the workspace, and where users are currently working. State transitions are triggered by actions performed by the users of the shared global workspace (such as modifying content of nodes or adding links). In addition, state transitions can also be triggered by time-dependent actions (such as reaching a deadline). In order to orient themselves in such a shared global workspace, which is used over a longer period of time, users need to find out about the past, present and future of the workspace:

1. Comprehending the past (history) of the shared global workspace requires understanding the current state of the global workspace (in terms of structure and content) and understanding the changes and sometimes also why and when and by whom they occurred.
2. To be able to coordinate activities with others one must be able to assess the present, i.e. the current state of the shared global workspace (who is doing what right now).
3. To be able to recognize conflicts and opportunities for synergy requires being able to make informed decisions about the future, i.e. what to do next. This requires knowledge about current tasks to be performed in the group and about relevant parts of the shared global workspace.

Answers to these three groups of issues need to take into account the dynamic nature of cooperative work in a shared global workspace. Tasks and plans might change or even evolve in the course of collaborative work, and opportunities for synergy and conflict usually develop unplanned and in unforeseeable ways.

Also, not all aspects are equally important to users all the time. When making a decision about which task to work on next, users need knowledge about the current state (in terms of who is currently working on which tasks) and about the planned tasks for the future. However, when working on a task, users are more likely to need knowledge about people working on tasks or objects "relevant" to their own work. Thus, they may need a different kind of overview of the shared global workspace than they would need for working on a new task. This general problem can be considered to be a problem of dealing with information overload.

Coordination in such a shared workspace, where teams create and manipulate shared artifacts to solve a complex problem (e.g. to develop software, to design a product, or to write a research paper), needs to deal with:

- Work planning:
  Here, planning of how to achieve the common goal (e.g. by creating tasks, relating them to each other and to deliverables and schedules) is important. As experience shows, task planning cannot be done once for all at the beginning. Rather, it is an ongoing activity, where the team continuously refines the task plan and adjusts it to changes in the environment and according to the progress of knowledge in the team. Task planning is a way of achieving coordination between team members by agreeing on how to split and integrate individual work.
- Work execution:
  Here, coordination is needed when actually carrying out the work. This does not only address the need of selecting tasks to be worked on in a meaningful

way, but also addresses the need for coordination between people working on related tasks or working on the same objects/resources in different tasks. These dependencies partially arise at run-time. They should not only be seen as a threat to consistent work, but also as an opportunity for achieving synergy in the team.
- Work monitoring:
  In work monitoring the task is to get an overview about the "state of affairs". Are tasks progressing as planned? Do things fit? Is there a need to change the plan?
- Consistency management:
  This may be seen as a part of work monitoring (i.e. to maintain consistency between the work plan and the actual performance of the work). Alternatively, one can see consistency management as a general problem, which also addresses consistency between tasks and their results, and how work performance, work plan, and the overall goals relate to each other.

Overall, current approaches to implement shared workspaces only address some of these issues.

Previous work focused either on providing means for coordination or on supporting overviews. Approaches to coordination in shared workspaces include the provision of group awareness, intra- and inter group conventions [12], documentation of previous states (as in versioning systems, or using log files such as in NoteCards [22]), process support (such as shared plans or workflow management) or communication support. Means for retaining overviews include fisheye views, flexible Diff-ing [14] or shared task lists. However, previous approaches do not address all the requirements of supporting overviews in shared workspaces.

3 Previous Approaches

Coordination in shared global workspaces is a known problem in the CSCW area. Previous approaches include

- provision of group awareness in shared workspaces,
- provision of direct communication channels between collaborators (e.g. notifications, messaging services, informal communication via computer-supported conferences), and
- provision of explicit process support (e.g. workflow management, task-based versioning).

Group awareness [4] facilitates the assessment of the present state (who is doing what) in a shared workspace. It is usually a concept that is applied to shared local sub workspaces in order to help synchronously cooperating users to coordinate their activities. It also supports - to some degree - making informed decisions about what to do next (based on knowledge about who is working in which part of the shared local workspace, thus showing some opportunities for synergy or conflict). However, it does not support finding out about the past, and it does not explicitly address current tasks and plans of the group. Most systems in this area provide local awareness [9]. Examples are applications implemented in GroupKit [19] and Suite [3] as well as SEPIA [21], DOLPHIN [20], TeamRooms [18], and ShrEdit [11]. These examples used user lists, activity markers, shared views, and telepointers. Some other systems also provided tools for global awareness [9]. Examples include task lists as in CoAUTHOR [10] or workflow management systems, radar views as in SEPIA, or history logs as in NoteCards [22]. BSCW [1] provides workspace awareness through a user presence and activity monitor since version 3.2. This tool shows the presence of participating users in the workspace. It also facilitates communication among users. In addition, BSCW provides access control, simple versioning of documents in a workspace, and asynchronous workspace reports. ORBIT [13] organizes a shared workspace using the locales concept (as a means to communicate tailored awareness) and supports global awareness on other locales through a "navigator". The RICH system provided hierarchical view filters as a means for searching hypermedia networks for content and change related predicates [23]. However, RICH does not support search for collaboration related predicates. Another interesting approach is to provide activity awareness between different individual workspaces as in the Interlocus system [15]. Here, notifications and awareness functions provide asynchronous workspace awareness. However, synchronous awareness and assessment of future activities are not supported. Finally, GroupDesk [5] provides presence awareness and activity awareness in a shared workspace. It uses the concepts of work situation and interest context to distribute events in the shared workspace.

Direct communication channels may be used to provide some group awareness (e.g. informing collaborators about ongoing activities via notifications) and may also help to assess the current state and potentials for synergy and conflict. An example is NSTP [2], which provides an infrastructure e.g. to share meta-information about a shared workspace. Here, workspace structure and content are described as places, things, and facades, to enable sharing and navigation in the shared workspace. However, NSTP itself does not offer awareness functions to applications. Using informal communication collaborators may determine which tasks or parts of the document they should work on next. However, assessment of past changes and finding out about parts of the shared global workspace relevant to the task at hand is not supported.

Explicit process support such as WFM type systems (e.g. GroupDesk [5]) and task-based versioning (e.g. CoVER [6] and Verse [7]) facilitate informed decisions about the future (based on currently active tasks, or tasks to be activated next). Also, the past can be assessed (in terms of activities finished or active, their states, and the respective versions of the shared global workspace). However, these systems require detailed planning in advance and usually cannot deal very well with frequent changes and emerging structures. In addition, finding out about relevant parts of the shared global workspace, which were not initially modeled as part of the process structure, is not supported. Also, these systems largely focus on asynchronous collaboration (with the exception of the versionedSEPIA system [6]).

As a result of this analysis we can state that previous approaches do not address all the aspects of coordination in shared global hypermedia workspaces. In the next section the potential of structural computing for addressing the coordination problem is examined.

4 Supporting Coordination by Structural Computing?

An interesting possibility of supporting coordination in such a shared workspace arises when looking at the coordination problem from the structural computing point of view. In this case, one could view the shared workspace as a medium for collaboration and coordination. The shared workspace may

1. capture shared artifacts describing objectives as well as plans and arguments of how to achieve the final result (i.e. address work planning).
2. support actual performance of work by providing team members with access to tasks, to their related information resources and tools, and to information about ongoing work in other tasks, which might be of potential interest to the current task (i.e. address work execution). This can even include active suggestions about which people to contact for coordination purposes. Obviously, supporting work execution requires integration of all resources required for tasks. For this reason, the open hypermedia systems approach [17] seems highly useful in this domain.
3. enable monitoring of work by providing access to overviews about completed and active tasks, and their relationship to the work plan.
4. support consistency between the components of the shared workspace.

When developing such a shared workspace environment, which could provide the above functionality, the main problem lies in the interdependency between

- the actual content of the workspace (e.g. objectives, plans, task descriptions, and results, as well as the description of the team including members, roles, and organizational hierarchies) and
- the coordination behavior necessary to let the collaborative work progress smoothly.

Both aspects influence each other, and are highly fluent (i.e. changing continuously). Plans might trigger users to do certain actions, actions of co-workers change the state of the shared workspace, and these changes might require different coordination behavior. Here, coordination can aim at the global level or at the local level:

- Coordination on the global level (global coordination) can be defined as the task of minimizing differences between the actual state of the workspace and the terminal state to be achieved over a number of steps (occurring over the lifetime of the collaboration). Constraints such as "all deliverables have been produced", "deadlines have been met", etc may characterize the terminal state.

- Coordination on the local level (local coordination) aims at aligning activities of team members working on different parts of the project (e.g. on different tasks, on different documents). Some needs for local coordination may arise from the global coordination need (e.g. caused by dependencies between tasks), while other needs for local coordination may arise from user behavior (such as navigating in the shared workspace, or having special expertise).

So, how can we build such a system? Clearly, the open hypermedia systems approach [17] can help to easily integrate information. However, it does not support the continuous change of coordination support adapted to the situation at hand. It is here were structural computing can potentially help.

Figure 1 shows the three main components of the shared global workspace, which have been proven useful to facilitate coordination:

- Content structure,
- Team structure, and
- Process structure.

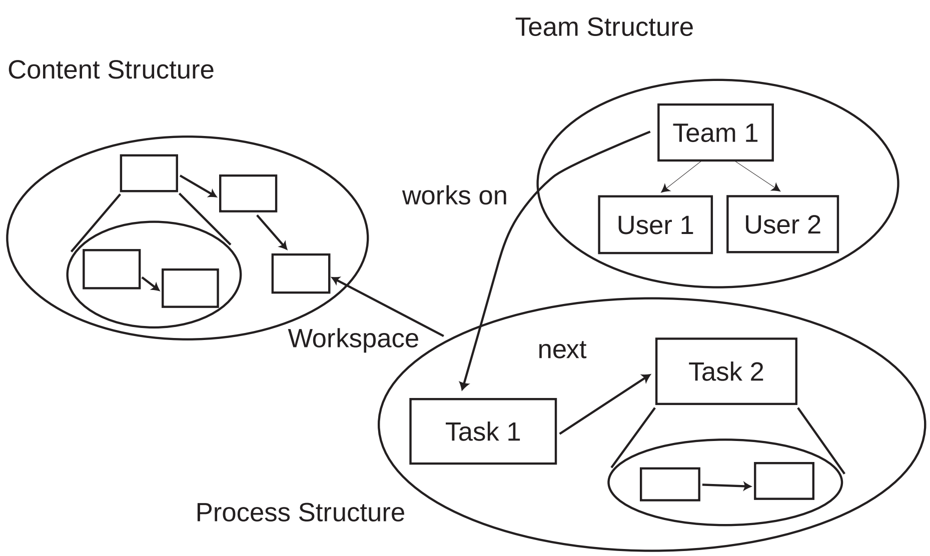

Fig. 1. Example components of the shared workspace; nesting of composite nodes is indicated by showing dashed lines between nodes (indicated by rectangles) and the subnetwork they contain (indicated by ovals).

The content structure models the content of the shared global workspace (i.e. the artifacts and their relationships) using typed hypermedia nodes (atomic and composite nodes) and hypermedia links. Different node and link types are supported, which can be used to express domain semantics as well as constraints. Using an extensible type system ensures that emerging task-oriented structures can be supported. For more details see [8].

The team structure models users and teams working in the shared global workspace using again hypermedia objects and representing relationships between users, teams and artifacts using typed hypermedia links. Examples for such relationships are teams being composed of users, users owning pages, and teams working on pages. This dual use of hypermedia supports simple editing and browsing of organizational as well as content structures in the global workspace. Note that content and team structures are connected via relationships (such as, a user owns a node). These relationships can be represented as links.

The process structure models tasks and activities performed in the shared global workspace. It uses nested hypermedia nodes (of type "task") and represents dependencies (such as temporal order, control flow and data flow among tasks) between them using hypermedia links. Again, the process structure is connected with the content structure (e.g. a task uses a node as its work area) and the team structure (e.g. a task is performed by a team or a user).

In our approach we assume that the shared global workspace is in principle available to all group members all of the time. This requires means for synchronous access to the shared global hypermedia workspace such as, e.g., implemented in the DOLPHIN [20] and the CHIPS [8] system. However, since group members may join and leave the workspace asynchronous work can also be supported.

Since all three aspects of the shared global workspace (its content, its users, the way it is used) are represented in the shared hypermedia workspace, tools can be provided that exploit the joint data model to facilitate orientation and coordination (for more details on these tools, cf. [9]).

The open issue is now how to support the adaptation between the content of the shared hypermedia workspace and the coordination functionality. Letting the structure of the workspace and the structure of the activities being executed on that workspace drive the coordination support might actually devise a new approach to coordination of shared work. In this approach, the structures present in the shared workspace are regarded as building blocks, on which behaviors (i.e. computations over structure) can be used to express structural semantics (i.e. coordination behavior matching the needs of the structure). Potential structures in a shared hypermedia workspace include:

- Process structure:
  Task networks (describing pieces of work to be done as well as flows of control and information) can be used to define coordination behavior aimed at presenting users with information regarding the context of their work, including:
  - Task structure (predecessor-successor tasks, work decomposition hierarchy, and previous results which should be reflected in their work)
  - Concurrent activities in related tasks

– Team Structure
  Teams composed of users carrying roles reflect organizational dependencies. The connection between team structure and process structure can be used to define coordination behavior aimed at presenting users with information regarding e.g.
  - Responsibility of teams and users
  - Competencies of teams and users (which may be deduced from previous tasks that have been tackled by the users)
– Content structure
  The content structure models the content of the shared global workspace (i.e. the artifacts and their relationships) using typed hypermedia nodes (atomic and composite nodes) and hypermedia links. Using the relationships between process structure, team structure, and content structure coordination behavior can be defined, which aims at presenting users with information regarding e.g.
  - Other tasks related to the same document
  - Other people related to the same document
  - Other documents, which may be related due to their use in related tasks or teams.

Now, structural computing could help to facilitate coordination by deriving the required coordination functionality from the above structures, which are present in the shared workspace. Local coordination could be supported by behaviors over the above three structures, which would present the above kinds of information to users in awareness or coordination tools (which need to be provided on the shared workspace) or as part of existing tools to manipulate these structures in the shared workspace. The latter would require including special awareness or coordination widgets.

Defining behaviors that test and reflect the dependencies between objectives and planned tasks (as part of the process structure) and content structure (respectively the content of the tasks being executed) could support global coordination. However, at this point it is unclear how these constraints could be defined (e.g. as static rules or as meta-rules defining how the actual behaviors are computed from the structures present in the shared workspace).

Implementing above functionality define some requirements for structural computing:

– The respective computations need to be defined, which derive the needed coordination behavior from the hypermedia structures in the shared workspace. Such coordination behavior requires (1) the selection of the appropriate user interface elements or tools (on the appropriate objects), and (2) the creation of new hypermedia objects in the shared workspace, which provide derived information (e.g. new versions, references to interesting objects, notifications of relevant users).

- Coordination behavior depends on the teams and users of the shared workspace, and their preferences etc. Thus, these aspects need to be modeled as structures in the shared workspace, too, so that they can influence the coordination behavior exhibited by the workspace.
- Since coordination behavior also depends on current use, information such as the users currently logged in, the sessions they work in collaboratively etc. must be represented as structures in the shared workspace, too.

5 Conclusions

In this paper, properties of the coordination problem in shared hypermedia workspaces were analyzed. Based on the deficits of existing approaches to local and global coordination, the potential of using a structural computing approach was discussed. The main benefit of such a structural computing approach, which explicitly models processes, teams, and content as part of a shared workspace, is that behavior (i.e. computations over structure) can be used to dynamically define coordination support.

However, up to this point there exist no implementation of such an approach. Thus, next steps include a proof-of-concept implementation of the approach as well as design considerations for the user interface of such a system. Application domains for this approach include extended enterprise engineering, a discipline where independent companies combine their resources (i.e. including their work processes) to deliver new products in a short time frame. Here, process structure and coordination requirements cannot be defined up front, and they are likely to change fast over the execution of such a project.

6 Acknowledgements

Many thanks are due to Ken Anderson for his detailed comments on earlier versions of this paper.

References

1. Bentley, R., Horstmann, T., Trevor, J., The World Wide Web as enabling technology for CSCW: The case of BSCW, Computer-Supported Cooperative Work: Special issue on CSCW and the Web, Vol. 6 (1997), Kluwer Academic Press.
2. Day, M., Patterson, J. F., Mitchell, D. The Notification Service Transfer Protocol (NSTP): Infrastructure for Synchronous Groupware. Proc. of 6th International WWW Conference, April 7-11, 1997, Santa Clara, USA. http://atlanta.cs.nchu.edu.tw/www/PAPER80.html
3. Dewan, P, Choudhary, R. Flexible User-Interface Coupling in Collaborative Systems. In Proc. of ACM CHI'91 Conference, April 1991, pp. 41-49.
4. Dourish, P. and Bellotti, V. Awareness and Coordination in Shared Workspaces. In Proc. of the ACM 1992 Conference on Computer Supported Cooperative Work, Toronto, Nov. 1-4, pp. 107-114.

5. Fuchs, L., Pankoke-Babatz, U., Prinz, W. Supporting Cooperative Awareness with Local Event Mechanisms: The GroupDesk System. In Proc. of the 4th European Conf. on CSCW (ECSCW'95), Stockholm, September 10-14, 1995, pp. 247-262.
6. Haake, A., Haake, J. Take CoVer: Exploiting Version Support in Cooperative Systems. In Proc. of InterCHI'93, Amsterdam, April 26-29, pp. 406-413.
7. Haake, A., Hicks, D.: VerSE: Towards Hypertext Versioning Styles. In Proc. of 7th ACM Conference on Hypertext, Washington, March 16-20, pp. 224-234.
8. Haake, J. M., and Wang, W. Flexible support for business Processes: Extending cooperative hypermedia with process support, Proceedings of Group'97, Phoenix, November 16-19, 1997, pp. 341-350.
9. Haake, J. M. Facilitating Orientation in Shared Hypermedia Workspaces. In: Stephen C. Hayne (Ed.): Group'99. Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (November 14-17, 1999, Embassy Suites Hotel, Phoenix, Arizona, USA), pp. 365-374, New York, ACM Press, 1999.
10. Hahn, U., Jarke, M., Kreplin, K, Farusi, M. CoAUTHOR: A hypermedia group authoring environment. In Proc. of the 1st European Conference on Computer Supported Cooperative Work (EC-CSCW'89), Gatwick, U.K., September 13-15. Computer Science House: Sloug.
11. MacGuffin, L. Olson, G. M. ShrEdit: A Shared Electronic Workspace. Technical Report No. 45, University of Michigan, Cognitive Sciences and Machine Intelligence Laboratory. 1992.
12. Mark, G. Merging Multiple Perspectives in Groupware Use: Intra- and Intergroup Conventions. Proceedings of Group'97, Phoenix, November 16-19, 1997, pp. 19-28.
13. Tim Mansfield, Simon Kaplan, Geraldine Fitzpatrick, Ted Phelps, Mark Fitzpatrick, Richard Taylor. Evolving Orbit: a progress report on building locales. Proc. of Group'97, ACM Press, Phoenix, AZ, Nov 1997.
14. Neuwirth, C. M., Chandhok, R., Kaufer, D. S., Erion, P., Morris, J., Miller, D. Flexible Diff-ing in a Collaborative Writing System. In Proc. of the ACM 1992 Conference on Computer Supported Cooperative Work , Toronto, October 31-November 4, pp. 147-154.
15. Nomura, T., Hayashi, K., Hazama, T., Gudmundson, S. Interlocus: Workspace Configuration Mechanisms for Activity Awareness. In Proc. of the ACM 1998 Conference on Computer Supported Cooperative Work (CSCW'98), Seattle, Washington, November 14-18, 1998, pp. 19-28.
16. Nürnberg, P. J., Leggett, J. J., Schneider, E. R. As we should have thought. Proceedings of ACM HT'97, pp. 96-101.
17. Reich, S., Wiil, U. K., Nrnberg, P. J., Davis, H. C., Groenbaeck, K., Anderson, K. M., Millard, D. E., Haake, J. M. Addressing Interoperability in Open Hypermedia: The Design of the Open Hypermedia Protocol. The New Review of Hypermedia and Multimedia, Taylor Graham Publishing: London. January 2000.
18. Roseman, M., Greenberg, S. TeamRooms: Network Places for Collaboration. In Proceedings of the ACM CSCW '96, Boston, November 1996, pp. 325-333.
19. Roseman, M., Greenberg, S. GroupKit: A Groupware Toolkit for Building Real-Time Conferencing Applications. In Proc. of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, October 31-November 4, pp. 43-50.
20. Streitz, N., Geissler, J., Haake, J., and Hol, J. DOLPHIN: Integrated meeting support across local and remote desktop environments and liveboards. In Proc. of ACM CSCW'94 (October 1994), pp. 345-358.

21. Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schütt, H., and Thüring, M. SEPIA: a cooperative hypermedia authoring environment. In Proceedings of ACM Hypertext'92 (1992), pp. 11-22.
22. Trigg, R., Suchman, L., Halasz, F. Supporting Collaboration in NoteCards. In Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW'86), Austin, December 3-5. Pp. 153-162.
23. Wang, W. and Rada, R. Structured Hypertext with Domain Semantics, ACM Transactions on Information Systems, Vol. 16, No. 4 (Oct. 1998), pp. 372-412.