Transaction Support for Cooperative Hypermedia Document Authoring
— A Study on Requirements —

Thomas Tesch & Jürgen Wäsch

GMD-IPSI Integrated Publication and Information Systems Institute
Dolivostraße 15, D-64293 Darmstadt, Germany
{tesch,waesch}@darmstadt.gmd.de

Abstract

Non-standard database applications like CAD, CASE, or cooperative authoring systems impose new requirements on database management systems (DBMS). While object-oriented DBMS provide appropriate data modelling facilities they still lack adequate transaction management support for this kind of applications. Cooperative authoring processes are mainly characterized by long duration activities that are interactively performed by multiple authors. They cooperate on shared documents to produce a final, commonly accepted, hyperdocument.

Thus, a transaction model that is well-suited for such an environment has to support long-running, interactively controlled activities, in which the competition for resources is replaced by the need to cooperate. In this paper, we present requirements for a cooperative transaction model to overcome the limitations of the traditional transaction model. The requirements are derived from the analysis of the cooperative hypermedia document authoring domain.

1 Introduction

Non-standard database applications like CAD, CASE, or cooperative authoring systems impose new requirements on database management systems (DBMS). While object-oriented DBMS provide appropriate data modelling facilities they still lack adequate transaction management support for this kind of applications. Furthermore, current DBMS or office information systems provide only minimal support for user-to-user interactions. This kind of support is required for activities like authoring where a significant portion of work is performed in the context of a group.

Cooperative authoring processes are mainly characterized by long duration activities that are interactively performed by multiple authors. They cooperate on shared documents to produce a final, commonly accepted, hyperdocument.

Thus, a transaction model that is well-suited for such an environment has to support long-running, interactively controlled activities, in which the competition for resources is replaced by the need to cooperate. The emphasis, therefore, is not on preventing access to resources, but rather on the semantically correct exchange of information among concurrent activities of cooperating users. At the same time, we are interested in preserving selected transactional properties of activities performed during a cooperative authoring process.

The goal of the ESPRIT III project TransCoop[1] is the development of a cooperative transaction model and a corresponding specification language that are applicable for a wide spectrum of cooperative applications. Beside the cooperative authoring domain, we analyzed Design for Manufacturing [VFSE95] and workflow applications [JLP$^+$95] to derive requirements for a cooperative transaction model and a corresponding specification language [TV95, VT95].

The paper is organized as follows. In section 2, we discuss the related work in the area of computer supported cooperative work (CSCW) and advanced transaction models. Afterwards, we describe the cooperative hypermedia document authoring domain in section 3. From this analysis we derive requirements for a cooperative transaction model in section 4. Finally, we present conclusions and further work.

2 Related Work

Related work addressing the synchronization of multiple collaborating users can be found in the field of CSCW [WL93, GS87] and advanced transaction models [Elm92].

The former approaches follow in most cases a more or less ad-hoc approach in synchronizing the work of collaborating users, e.g., user-controlled locking of objects without transactional support, whereas the latter are based on the transaction concept [Gra81].

Conservative approaches follow the check-in/check-out model which is a commonly accepted approach to synchronize access to shared data objects [BKK85, KKB87, KSUW85]. Users explicitely copy objects from a shared database to private workspaces to manipulate them (check-out). Check-out objects are reserved for exclusive access until a later check-in such that they are not accessible for other users. This may result in blocking of designers over long periods of time which is not acceptable in cooperative authoring. If this approach is combined with versioning mechanisms, new object versions can be derived concurrently but they need later to be merged manually by the users.

[GS87, KP90] introduced the notion of a floor to synchronize multiple users working with a set of shared objects. The floor is owned by one user at a time who has the exclusive permission to execute all modifications on the objects while other users can only observe his actions by means of read operations. A floor passing protocol allows to establish a cooperative behavior by exchanging the floor between different users. Usually a single floor is used for all objects which reduces the possible degree of concurrency. Availability of data is limited and, the major disadvantage, the problem of ensuring correctness is delegated to the users, i.e., they are responsible for leaving the database in a consistent state.

Another alternative is not to provide any system support to synchronize concurrent operations of co-workers at all. Social protocols leave the responsibility for correct and consistent executions at the users. Instead of synchronization mechanisms they provide advanced coordination facilities, e.g., communication channels like audio and video, that

[1] This work is partially done in the ESPRIT III BRA project TransCoop (EP8012) which is funded by the Commission of the European Communities. The partners in the TransCoop project are GMD (Germany), University of Twente (The Netherlands), and VTT (Finland).

allow for explicit coordination of the involved users. Systems using social protocols are described in [EGR91, SFB+87]. Because correctness and failure atomicity guarantees have to be ensured by the users, this approach can result in several conflicting operations that are not detected by the system. Useful pieces of work may need to be undone or have to be repeated if the authors are not aware of conflicts.

The approaches discussed above do not meet the synchronization requirements of cooperative authoring environments (and CSCW applications in general) because they give no correctness guarantees in the transactional sense.

Transaction models in general are guaranteeing fault tolerance for a specified granularity and synchronize concurrent activities of multiple users while ensuring build-in correctness criteria. In cooperative authoring environments, groups of users are working on large collections of documents, sharing their contents. Furthermore, authoring activities usually last over a long period of time. Hence, approaches not addressing a failure resilient execution are not feasible because system crashes and deadlocks affect long-lived activities more often than short ones.

The traditional ACID transaction model [Gra81, HR83] does not satisfy the needs of cooperative applications. It assumes a lot of short transactions consisting of predefined sequences of read/write operations. ACID transactions give users the illusion of a *single* user system by enforcing equivalence to a serial execution of transactions — this contradicts the needs of cooperative applications.

Transaction models addressing cooperation aspects support a group of users engaged in a common task. Cooperative software engineering, engineering design and cooperative authoring are typical applications of these models. Well known approaches in this area are split-join transactions [PKH88] and transaction groups [FZ89].

The split- and join-transaction approach [PKH88] supports the restructuring of transactions. A split-operation allows to split a running transaction into two new (serial or independent) transactions while a join-operation allows to incorporate two independent transactions into a new transaction. This model allows to establish a cooperative behavior by exchanging parts of transactions between concurrent users. Since the model supports only single data copies, the availability of data is limited if a locking protocol is used. Another disadvantage is that no higher level operation semantics are taken into account.

Group-oriented transaction approaches describe the overall working process as a transaction hierarchy consisting of group transactions. Individual user transactions form the leaves of the transaction hierarchy. Visibility between transactions is supported by extended lock schemes [KSUW85] or by following predefined access patterns that define the application-specific correctness criteria [FZ89]. The usability of this approach is restricted because significant parts of the cooperative application have to be pre-specified in order to describe whether a particular non-serializable execution is correct.

3 Cooperative Hypermedia Document Authoring

In the following, we describe the cooperative authoring process considering the specific characteristics of hypermedia documents. First of all, we focus on the specific structure of hypermedia documents in contrast to traditional documents. Afterwards we describe the authoring process emphasizing coordination and cooperation aspects when multiple authors are working together. However, most of our observations are valid for traditional authoring processes as well.

3.1 Hypermedia Documents

Hyperdocuments form a new class of (electronic) documents, that is based on the concept of hypertext [Nel81]. A hypertext is a collection of information entities (nodes) containing cross-references (links) which — with the aid of an interactive browser program — allow the reader to navigate easily within the document.

The main difference to traditional (paper) documents is that hyperdocuments are non-linear. The concept of links gives authors the ability to connect information entities, such as text nodes, in an almost arbitrary way. Therefore, a hyperdocument has an associative, network-like structure.

An important implication of the hypertext concept is the interactive usage of hypertexts by their readers. A reader can freely navigate through the hypertext structure because links are represented explicitly and therefore references can be interpreted automatically by a computer system. Thus, there are no restrictions imposed by the presentation medium, i.e., the reader can decide which link to follow. Of course, authors can create some predefined tours through the hypertext. Such guided tours act as guidelines for the exploration of the hypertext by the reader.

The nodes and links which form a hyperdocument can be stored in an ordinary file system or can be mapped onto database objects in a database management system. Applications may provide various presentation styles supporting the authoring process or addressing different readerships, respectively.

If the nodes of a hyperdocument do contain (besides text and simple graphics) also multimedia information, such as complex pictures and continuous information (audio, video, animation), the concept of hypertext is generalized to hypermedia. The term hypermedia addresses multimedia extensions of hyperdocuments, whereas the term hypertext is more related to the structural aspects of hyperdocuments.

There are several extensions of the basic hypertext model, such as the introduction of composite nodes [Hal88] and atomic nodes. Composite nodes introduce the concept of aggregation into hypertext structures. They can contain other links and nodes and can be used to cluster related information entities. Atomic nodes represent single information entities.

3.2 The Authoring Process

In this section we give a deeper understanding of the authoring process of hypermedia documents. We focus on those parts that have significant impact on the design of a cooperative transaction model. Hence, we do not present recent results from writing research that do not have direct impacts on the authoring process from a transactional point of view.

When describing the cooperative authoring domain we have to contemplate two different aspects. The product aspect comprises certain constraints and requirements posed on the final product hyperdocument which is the result of the authoring process. On the other hand, the process aspect focuses on the concrete authoring activities, i.e., the production of an high quality artifact. Both aspects have several implications when thinking about a transaction model suitable for cooperative authoring.

The process of writing hyperdocuments is regarded in literature as a problem solving process [HF86] or, more specific, as a design activity [THS90, HT93]. A characteristic of design activities is that the problems to be solved are not clearly defined. This is reflected by a problem definition which is marked by gaps that are to be filled. The understanding of those gaps is a significant part of the problem solving process. Thus, the analysis of the problem itself is often the most important step for the solution [Sim81]. The weak problem definition imposes further uncertainty on the problem solving process, i.e., the problem solving process usually can not be specified in beforehand.

This can be easily illustrated by considering how documents are designed. At the beginning, there is only some weak knowledge about the final document, e.g., its subject and the target group of readers. A few constraints of the final document may be available, e.g., size limitations or prescribed layouts. One could start outlining the structure of the document or just collecting existing ideas in arbitrary way or start with studying further literature.

[HT93] decomposes the overall design problem of writing (hyper)documents in three subproblems addressing the planning of the process itself, i.e., development of a basic document structure and a workplan (planning problem), the acquisition of background material (content problem), and the expression of the authors intentions in a way suitable for the aimed readership (rhetorical problem).

Beside these subproblems [SHT89] identifies a fourth subproblem for argumentative documents referring to the construction of argumentation chains (argumentation problem). Although this decomposition of the overall problem in more well-defined subproblems reflects a hierarchical structure and eases the problem solving process, it is important to notice that the identified subproblems are highly interdependent.

The identified subproblems of the authoring process are mapped in the SEPIA system into different authoring activities [HW92]. Hence, in accordance to the presented model, a planning activity, a content activity, a rhetorical activity, and an argumentation activity were identified within SEPIA.

Within the different authoring activities, authors choose interactively their actions and frequently undo/redo them as they go along, i.e., the exact sequence of actions within an activity cannot be determined in advance like in office procedures with a more or less predefined set of organizational rules. Hence, authoring systems have to consider this opportunistic nature of the authoring process. They should not restrict the authors to follow a predefined sequence of actions.

Authoring processes are usually of long duration. One reason for this is their opportunistic nature and the lack of a predefined problem solving strategy. Because for authoring processes no *optimal* solution can be found [Sim81], authors always have the impression that the current solution could be improved. This embarrasses authors from finishing the process and, hence, authors have to compromise to a product that is "good enough", often due to deadlines.

3.3 The Cooperative Authoring Process

A significant number of documents result from the work of more than one author. Hence, authors should be assisted in designing the document in a cooperative fashion. Because we treat the authoring process from a database point of view, we assume a scenario in which multiple authors are manipulating a collection of shared documents that are stored in a database management system. The document structure consisting of nodes and links and the corresponding operations for manipulating the hyperstructures are modeled explicitely by means of the provided DML (e.g., VML in [WA95]).

A prerequisite for the cooperative authoring process is that the authors agree on common objectives of their authoring process to produce a single commonly accepted hyperdocument. The generation of a commonly agreed document representing the product "hyperdocument" is the result of a process of negotiations and cooperations between the involved authors. Hence, the authoring process performed by multiple authors can be considered as an extension of the problem solving process discussed in the previous section in which an additional subproblem occurs: the coordination problem [Haa94]. It refers to the coordination of authoring activities such that the synergy effect of cooperation can take place. The coordination of group activities strengthens their effectivity by reducing communication and collaboration effort and helps to avoid the occurrence of conflicting and repetitive actions. We already discussed that the nature of design processes like authoring contradicts a fully pre-planned coordination of activities. Hence, we have to accept that the appearance of some conflicts cannot be foreseen and, thus, the need to cooperate can arise spontaneously.

The essence of cooperation is the exchange of information between its participants. From the above discussion, it follows that cooperative authoring involves both, explicit exchange of information and implicit exchange of information through common accessible documents. The support of external communication is usually not part of the services of a cooperative transaction model. Having the design of a cooperative transaction model in mind, we concentrate on the latter kind of information exchange.

It follows that authors need to share partial results of their activities while still in progress. On the other hand, consistency of the shared artifacts has to be preserved. Sometimes a joint effort is necessary to transform the database into the next consistent state. In contrast to traditional transactions which assign consistent units of work to a single user, a consistent unit of work can be interactively composed by several interacting authors. Hence, it is necessary to coordinate the concurrent access to shared artifacts in a cooperative way such that effort is neither lost nor dublicated.

Depending on the characteristics of a specific cooperative authoring process, several cooperative situations may occur:

1. **Fully pre-planned cooperation:**

   This situation occurs only if the whole authoring process can be pre-specified. An example is the preparation of an electronic newspaper [S^+94] where the overall process can be arranged in several parts. These work parts are assigned to responsible co-workers and are performed sequentially if they depend on each other. It should be noted that a fine-grained specification describing all details of the cooperative authoring process can usually not be given in advance. The higher level activities may be structured but the concrete occurrences of low level actions cannot be forseen.

2. **Dynamic exchange of responsibilities:**

   If the cooperative authoring process cannot be fully pre-planned, authors may exchange their responsibilities during the process. By passing responsibilities, each author can gain the permission to change specific parts of the hyperdocument. This kind of cooperation does not allow simultaneous changes of the same document parts.

3. **Alternative versions:**

   In contrast to the situation described previously, it is now allowed that authors work simultaneously on the same parts of a hyperdocument by creating alternative versions of document parts. Work on alternative versions is done in isolation. Cooperation is achieved by merging the alternative versions into a new commonly accepted version of this document part. The merging process itself may be performed cooperatively.

4. **Synchronous work:**

   The closest kind of cooperation is established when authors are able to work simultaneously on the same document parts, e.g., as in joint editing scenarios. If no computer support is provided to synchronize the conflicts that may occur, co-authors must be careful in manipulating hyperdocuments. This cooperative situation occurs only in computerized authoring environments because in paper-based environments the physical sharing of documents between authors is not possible.

4 Transaction Support for Cooperative Hypermedia Document Authoring

With the traditional transactions, the user (transaction programmer and the end-user) is freed from concerns about partial (and therefore incorrect) results of transactions being left in the database in the case of a failure, or being observed by other users, concurrently accessing the system. These concerns are addressed by the transaction processing system that enforces commitment protocols ensuring transaction failure atomicity and concurrency control ensuring that the effect of concurrent execution of transactions is equivalent to that of running these transactions one at a time, without any interference from each other.

We already argued that these properties are not useful in cooperative authoring environments. In cooperative authoring environments failure atomicity is too strict and isolation prevents cooperation between co-authors. Thus, these properties have to be relieved.

Based on our analysis, we have derived the following requirements for a cooperative transaction model that is applicable to the cooperative hypermedia authoring domain. It turned out that most of the requirements are valid for Design for Manufacturing and workflow applications, too [TV95].

- **Relaxed atomicity:**

  The traditional ACID transaction models support the fault-tolerant execution of transactions by applying the all-or-nothing guarantee. Thus, all of a transaction's effects on the database are backed out in case of a failure or user-initiated abort.

  In opposite to this, the rollback of the whole cooperative authoring process in case of a failure is generally not acceptable. The amount of lost work should be minimized by a cooperative transaction model using mechanisms like savepoints or forward recovery [WR92].

  Due to the opportunistic nature of cooperative authoring processes, it is required that a cooperative activity should be able to proceed (and eventually succeed) even if other parts of the cooperative authoring process fail. A failure within an author's activity should not imply the rollback of another author's effort in their joint work.

  A cooperative transaction model should include appropriate mechanisms to react on different kinds of such exception situations. Thus, the specification of contingency activities, vital activities, and legal termination states etc. should be possible.

- **Relaxed isolation:**

  The sharing of final, as well as intermediate artifacts, e.g., drafts, among multiple co-authors is a prerequisite for the cooperative authoring process. Isolation of co-workers, as enforced in traditional ACID transaction models applying serializability as correctness criteria, contradicts the need of cooperation while the activities are in progress.

  Thus, the isolation property of a transaction model has to be relaxed in order to be applicable to cooperative authoring processes. Instead of isolation, a cooperative transaction model has to guarantee that no anomalies are introduced although tentative working results are exchanged among cooperating authors.

  Therefore, the model should offer appropriate primitives for the exchange of information between co-authors. These primitives have to ensure that this information exchange is semantically correct. The model should consider the semantics of activities and their actions occuring within an cooperative authoring process to enable a higher degree of cooperation between co-workers. Moreover, it should provide transactional services that support the spontaneous character of cooperation, e.g., primitives to delegate partial results and activities on demand of the authors.

- **Hierarchical decomposition:**

  The identification of different authoring activities within the overall authoring process reflects the hierarchical structure of authoring processes which are regarded as creative design processes. Moreover, due to permissions and responsibilities of authors in an organization and the structure of a specific authoring process, the work can be further decomposed.

  To support this hierarchical organization of cooperative authoring processes, the decomposition of the overall work into smaller subactivities need to be possible within a cooperative transaction model. The model has to offer appropriate language constructs to explicitly specify the activity structure. The mapping of the specification to corresponding transaction model primitives, like nesting, has to be possible.

- **Execution constraints:**

  To describe the pre-planned parts of the authoring process, a cooperative transaction model should allow the definition of execution constraints that describe the possible structure of an cooperative authoring process.

  This includes assignments of subactivities to co-authors and constraints on the occurrence of particular subactivities and on their execution order within the overall cooperative authoring process. These workflow-like constraints have to have the property to be runtime enforceable [ARSS93]. The transaction model should offer appropriate high-level language constructs for these execution constraints.

- **Retraction of decisions:**

  To reflect the opportunistic nature of authoring processes and to support the interactive user control of activities, a cooperative transaction model has to provide services that allow to retract decisions taken by the cooperating authors.

  Hence, the transaction model should support the compensation as well as the restart of authoring activities. Moreover, it should be possible to flexibly undo and redo actions within activities to allow authors to explore several alternatives to solve the problem.

- **Alternative versions:**

  To explore different alternatives of the same problem, different authors should be able to work at the *same* time on the *same* parts of the hyperdocuments.

  To avoid interference from co-workers, the cooperative transaction model should be able to manage alternative versions of objects. On demand of the authors, it should be possible to exchange alternatively derived versions and to combine them into a commonly accepted version.

Further requirements are posed on cooperative authoring systems. To support the mutual awareness of co-authors, notification mechanisms are required. Authors should be notified on significant events that are relevant for their current work and the further processing of the cooperative authoring process.

Moreover, additional communication facilities like e-mail, audio and video connections may be needed to allow co-authors to negotiate directly and coordinate their activities. Furthermore, the definition of social roles should be supported to assign privileges and responsibilities to authors. These services are not covered by a cooperative transaction model since it aims at synchronizing the cooperative access to persistent data.

5 Conclusion and further Work

In this paper, we presented requirements for a cooperative transaction model focusing on the cooperative authoring domain. As stated earlier, within the ESPRIT project TransCoop we investigated further cooperative application areas, namely Design for

Manufacturing [VFSE95] and workflow applications [JLP+95]l. The objective was to ensure that the transaction model is applicable for a wide range of cooperative applications.

The analysis of the three application areas was performed by investigation of current cooperative processes in industry , analysis of current software that supports cooperative work, and literature studies. We developed a taxonomy to compare the different application domains with respect to different categories. The complete analysis can be found in [TV95] and [VT95].

It has to be pointed out that the development of requirements is an ongoing process and, thus, the presented requirements may be extended or refined as we recognize new aspects of cooperation.

In the context of TransCoop we have proposed a cooperative transaction model that satisfies the above requirements [RKT+95]. We are currently working on detailed algorithms and a theoretical framework to prove the properties of the proposed model.

References

[ARSS93] P. C. Attie, M. Rusinkiewicz, A. Sheth, and M. P. Singh. Specifying and enforcing intertask dependencies. In *Proc. of the 19th Int. Conference on Very Large Databases*, pages 134–145, Dublin, Ireland, August 1993.

[BKK85] F. Bancilhon, W. Kim, and H. Korth. A model of CAD transactions. In *Proc. of the 11th Int. Conference on Very Large Databases*, pages 25–33, Stockholm, Sweden, August 1985.

[EGR91] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some issues and experiences. *Communications of the ACM*, 34(1):38–58, January 1991.

[Elm92] A. K. Elmagarmid, editor. *Database Transaction Models for Advanced Applications.* ACM Press. Morgan Kaufmann Publishers, Inc., 1992.

[FZ89] M. F. Fernandez and S. B. Zdonik. Transaction groups: A model for controlling cooperative transactions. In *Proc. of the Int. Workshop on Persistent Object Systems*, pages 341–350, January 1989. Newcastle, New South Wales.

[Gra81] J. Gray. The transaction concept: Virtues and limitations. In *Proc. of the seventh Int. Conference on Very Large Databases*, pages 144–154, 1981.

[GS87] I. Greif and S. Sarin. Data sharing in group work. *ACM Transactions on Office Information Systems*, 5(2):187–211, April 1987.

[Haa94] Jörg Haake. Autorensysteme für die kooperative erstellung von hypermedia–dokumenten. PHD Thesis, unpublished draft, 1994.

[Hal88] F.G. Halasz. Reflection on Notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7):836–852, July 1988.

[HF86] J.R. Hayes and L. Flowers. Writing research and the writer. *American Psychologist*, 41(10):1106–1113, 1986.

[HR83] T. Härder and A. Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317, 1983.

[HT93] J. Hannemann and M. Thüring. Schreiben als designproblem: Kognitive grundlagen einer hypertext autorenumgebung. *Kognitionswissenschaft*, Dezember 1993. Springer Verlag.

[HW92] J.M. Haake and B. Wilson. Supporting collaborative writing of hyperdocuments in SEPIA. In *Proc. of the ACM Conference on Computer-Supported Cooperative Work*, pages 138–146, Toronto, Canada, October 1992.

[JLP+95] J. Juopperi, A. Lehtola, O. Pihlajamaa, A. Sladek, and J. Veijalainen. Usability of some workflow products in an inter-organizational setting. In *Proc. of IFIP WG8.1 Working Conference on Information Systems for Decentralized Organizations*, Trondheim, Norway, August 1995.

[KKB87] H. F. Korth, W. Kim, and F. Bancilhon. On long-duration CAD transactions. *Information Systems*, 13, 1987.

[KP90] M.J. Knister and A. Prakask. DistEdit: A distributed toolkit for supporting multiple group editors. In *Proc. of the third ACM Conference on Computer-Supported Cooperative Work*, pages 343–355, 1990.

[KSUW85] P. Klahold, G. Schlageter, R. Unland, and W. Wilkes. A transaction model supporting complex applications in integrated information systems. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 388–401, Austin, Texas, USA, May 1985.

[Nel81] T. Nelson. *Literary Machines.* Mindful Press, CA, 1981.

[PKH88] C. Pu, G. E. Kaiser, and N. Hutchinson. Split–transactions for open–ended activities. In *Proc. of the 14th Int. Conference on Very Large Databases*, pages 26–37, Los Angeles, California, USA, August 1988.

[RKT+95] M. Rusinkiewicz, W. Klas, T. Tesch, J. Wäsch, and P. Muth. Towards a cooperative transaction model: The cooperative activity model. In *Proc. of the 21st Int. Conference on Very Large Databases*, September 1995. Zurich, Switzerland.

[S+94] K. Süllow et al. Multimedia forum - an interactive online journal. In Christoph Hüser, Wiebke Möhr, and Vincent Quint, editors, *Proceedings of Conference on Electronic Publishing*, pages 413–422. John Wiley & Sons, Ltd., April 1994.

[SFB+87] M. Stefik, G. Foster, D. Bobrow, K.Kahn, and L. Suchman. Beyond the Chalkboard – Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1):32–47, 1987.

[SHT89] N. A. Streitz, J. Hannemann, and M. Thüring. From ideas and arguments to hyperdocuments: Travalling through activity spaces. Arbeitspapiere der GMD 402, Gesellschaft für Mathematik und Datenverarbeitung, July 1989.

[Sim81] H. A. Simon. *The Sciences of the Artificial.* The MIT Press, Cambridge, MA, 1981.

[THS90] M. Thüring, J. Hannemann, and N. Streitz. Schreiben als Designproblem: Ein integrativer Ansatz. In D. Frey, editor, *Bericht über den 37. Kongress der Deutschen Gesellschaft für Psychologie, Göttingen*, 1990.

[TV95] T. Tesch and P. Verkoulen. Requirements for the TransCoop transaction model. Report TC/REP/GMD/D2-2/207, Esprit Project No. 8012, 1995.

[VFSE95] P. A. C. Verkoulen, F. J. Faase, A. W. Selders, and P. J. J. Oude Egberink. Requirements for an advanced database transaction model to support design for manufacturing. In *Proceedings of the Flexible Automation and Intelligent Manufacturing Conference*, pages 102–113. Begell House, Inc. New York - Wallingtord (U.K.), June 1995. Stuttgart, Germany.

[VT95] P. Verkoulen and T. Tesch. Requirements for the TransCoop specification language. Report TC/REP/UT/D2-1/014, Esprit Project No. 8012, 1995.

[WA95] J. Wäsch and K. Aberer. Flexible design and efficient implementation of a hypermedia document database system by tailoring semantic relationships. In *Proc. of the sixth IFIP Conference on Database Semantics*, May 30 – June 2 1995. Atlanta, Georgia.

[WL93] U. K. Wiil and J. J. Leggett. Concurrency control in collaborative hypertext systems. In *Proc. of the fifth ACM Conference on Hypertext*, pages 14–18, 1993. Seattle, Washington, Nov 14–18.

[WR92] H. Wächter and A. Reuter. The ConTract model. In Elmagarmid [Elm92], chapter 7, pages 219–264.