WISE: Business to Business E-Commerce

G. Alonso U. Fiedler C. Hagen A. Lazcano H. Schuldt N. Weiler
Swiss Federal Institute of Technology (ETH)
ETH Zentrum, Zürich CH-8092, Switzerland
E-mail: wise@ccic.ethz.ch
http://www.inf.ethz.ch/department/IS/iks/research/wise.html

July 31, 1998

1 Introduction

Business to business is one of the most successful forms of electronic commerce. Not long ago, it was only a prerogative of large enterprises with well established commercial links given the significant effort and investment required to deploy the necessary technology. The Internet and the proliferation of inexpensive computing power in the form of clusters of workstations or PCs has changed this situation entirely. Business to business electronic commerce is now affordable for many more enterprises which can collaborate by using the Internet to link their information processing systems. To complement this hardware infrastructure, the corresponding software support is now needed. Thus, one of the objectives of the WISE project (Workflow based Interned SErvices) is to develop and deploy the software infrastructure necessary to support business to business electronic commerce in the form of *virtual enterprises*. The idea is to combine the tools and services of different companies as building blocks of a higher level system in which a *process* acts as the blueprint for control and data flow within the virtual enterprise. From this idea, the final goal is to build the basic support for an *Internet trading community* where enterprises can join their services to provide added value processes. In particular, the project aims at designing, building, and testing basic infrastructure for business to business electronic commerce in the form of a working system capable of defining, enacting, and monitoring virtual enterprise business processes, as well as supporting related coordination activities (Figure 1). Such infrastructure includes an *Internet workflow engine* acting as the underlying distributed operating system controlling the execution of business processes, a process modeling tool for defining and monitoring the processes, a catalog tool for virtual enterprise services in which to find the building blocks for the processes, and a collaborative multimedia communication environment. The project also incorporates in its design considerations about security, quality of service, execution guarantees, exception handling, high availability, and scalability, as well as diverse other aspects related to WWW based interaction, catalog based information, catalog search, and communication frameworks.

In this short paper, we outline the main components of the project and describe the current status of the development efforts.

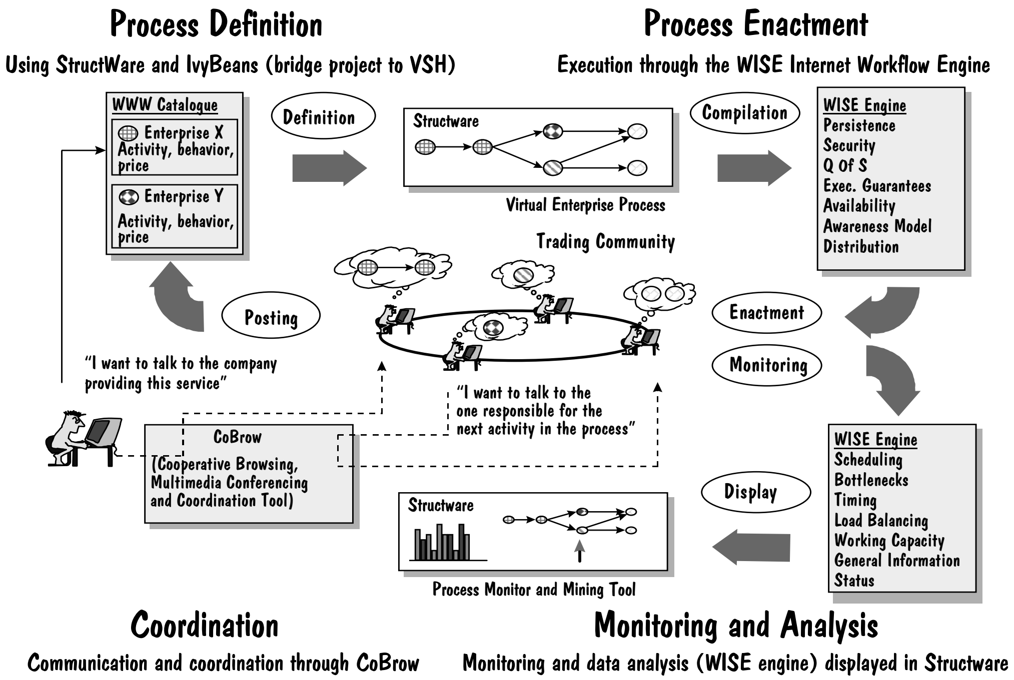

Figure 1: The different components of the WISE project

2 Project Goals

WISE is an integration effort in the sense that it tries to combine several different technologies into a coherent whole. This can be best seen by looking at the four main components (Figure 1) of the project. The first component is the *process definition*, which allows companies to make public the services they provide and allows designers to use these services to build virtual enterprise processes. We will refer to these companies as a *trading community*. The second component is the *process enactment*, which compiles the description of the virtual enterprise process and executes the process by invoking when appropriate the different services of the trading community. The third component is the *process monitoring and analysis tool*, in charge of observing and recording the execution of the virtual enterprise process. The data produced by the tool can then be used dynamically for load balancing and quality of service purposes as well as, later on, for analysis of the behaviour of the process. The fourth and final component is a *coordination and communication* mechanism that allows multimedia conferencing and cooperative browsing of relevant information among all participants in the execution of a process. This tool is context based in the sense that it understands the semantics of the process and establishes the communication patterns accordingly.

An important aspect of WISE is that these four components are tightly intergrated in the sense that they reflect an approach to electronic commerce based on transparency and ease of use but which can only be achieved if the functionality of the different components can be brought together. While the individual components are certainly important, the key to the success of WISE is the integration of the whole system.

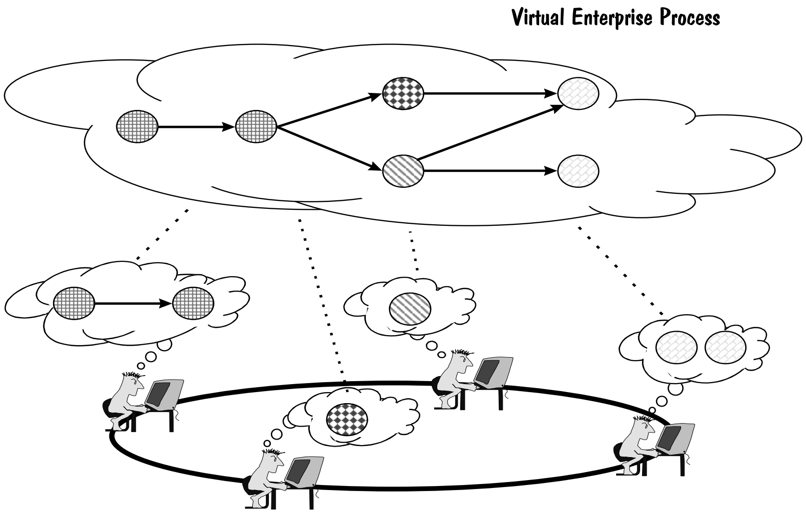

Figure 2: A workflow based virtual enterprise

2.1 Process definition

WISE will support virtual enterprise processes which are constructed by defining a process whose basic elements are the services of different companies (Figure 2). Each one of these services can be a process in itself although beyond providing the necessary interfaces, the nature of these services is orthogonal to WISE. The virtual enterprise process integrates the services of the different companies establishing the order of invocation, the control logic and the data flow between the participants in the same way a workflow process orchestrates business models within a single corporation. To make this idea a reality, there are two elements that WISE must provide. The first is a mechanism for the participants to publicize the services they provide. The second is a way to take this services and define a process based on them. For these purposes, WISE uses a WWW catalogue currently under development and a commercially available business processes modeling tool (Figure 3).

The WWW catalogue uses Java and active technology to allow companies not only to post their services to the catalogue but also to provide semantics to the entries in the ctalogue. Instead of having a list of mere "read-only" entries acting as pointers to the real service, the catalogue will contain a collection of objects, each one of them encapsulating all the data relevant to the corresponding service. With this approach, it will be possible to visualize the behavior of the service (for instance, if the service is a software component) and to display much more sophisticated information about its specification: cost, average duration, guarantees, requirements, etc.

From this catalogue, the virtual enterprise process designer can directly take the entries and use them as activities of a process modeling tool. The tool of choice is *Structware*, a product of IvyTeam, one of the partners in the project. Structware, which is internally based on Petri-nets, supports not only the definition of business models but also sophisticated analysis of its behaviour (bottlenecks, average execution times, costs, delays, what if analysis, etc.).

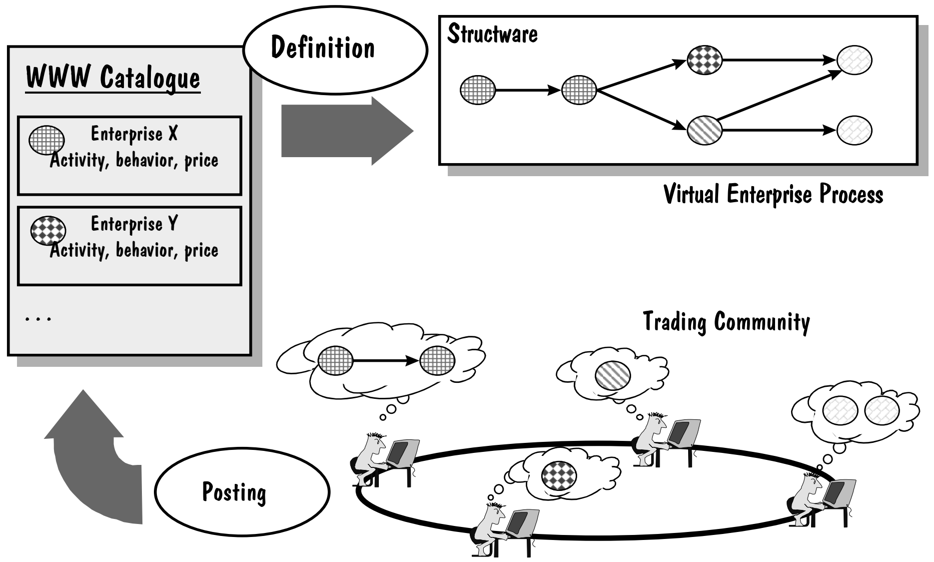

Figure 3: Process definition within WISE

In terms of process definition, Structware supports the standard flow control primitives of a workflow tool. It is possible to define conditional branching, nested processes, and assign additional information to each task within the process. This last point is extremely important from the point of view of WISE since it allows to use this additional information as the configuration information necessary to enact the process. In this way, no additional interfaces are necessary. The designer can define everything that is needed for the execution of the process using a single tool. Moreover, since the tool is only a simulation tool and not an enactment based tool, we can incorporate as well the activities posted to the catalogue without having to worry at this stage about how they will be invoked when the process needs to be executed. This separation of service definition and service enactment gives us the same advantages as the separation of interfaces and code in modular and object oriented programming. In fact, we see this entire procedure as a form of high level, coarse grained programming. We have successfully applied this idea of "workflow programming" within WISE and other projects in order to provide sophisticated language primitives that do not exist in other workflow tools. For instance, we can provide a very complete exception handling capability [HA98b] with characteristics that are very similar to that of conventional programming languages. Similarly, we have also developed an event handling mechanism that allows the introduction of asynchronous tasks in the control flow as well as enabling inter-process communication [HA98a]. This functionality is missing in current systems and we consider it is a crucial aspect to make a system like WISE succeed.

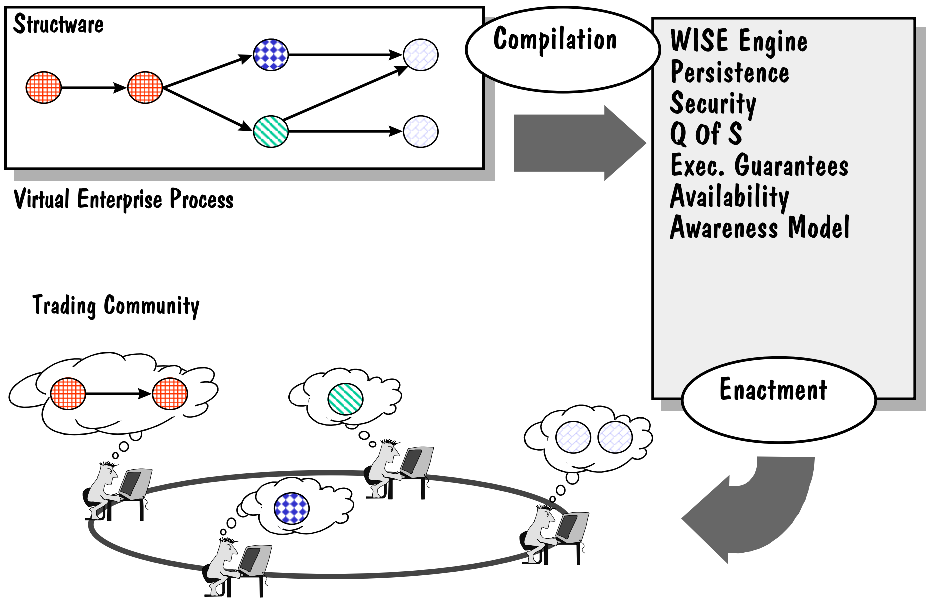

Figure 4: Process enactment within WISE

2.2 Process enactment

The enactment of the virtual enterprise processes is done by the WISE engine, which is based on work done within the OPERA project [Hag97, AHST97] which aims at developing a generic process support system which can be then tailored to particular applications. Within WISE, the OPERA engine will be extended in different directions (Figure 4). Of course, given the nature of the data exchanged between the different participants in the trading community, WISE will incorporate the necessary security mechanisms in the form of encryption of data for transmission over the network as well as a complete set of authentication measures for both execution, access, and monitoring of the processes. Also, to make the notion of trading community viable given the current limitations of bandwidth, the WISE engine will incorporate quality of service guarantees based on execution statistics and network characteristics. In this way, it will be possible to execute a given number of processes per time unit, thereby providing the necessary elements for the planning and deployment of the virtual enterprise. Another important aspect that the WISE engine will incorporate are execution guarantees, whereby a process will always be guaranteed to finish in a consistent state either by removing all changes it has introduced or by forcing it to terminate following a sequence of actions with a pre-determined outcome [SAS98]. Finally, WISE will also include an awareness model that will allow the engine to make decisions based on its own status and that of the participants. This awareness model is necessary for load balancing, increased availability, conflict resolution, notification mechanisms, and the handling of exceptions. It will also be used as the basis for context based communications in which a participant can request to exchange information with other participants based on the roles they have played during the execution of a given process.

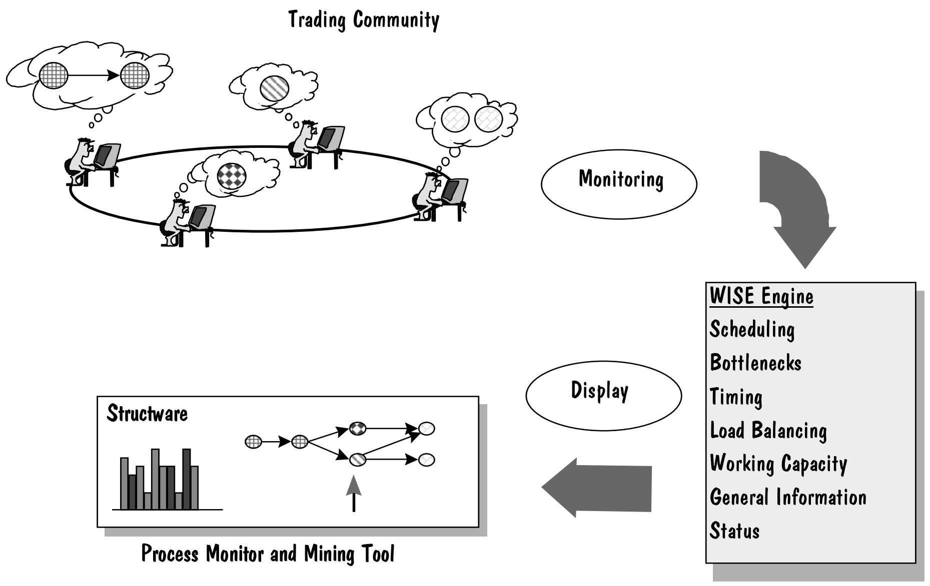

Figure 5: process monitoring within WISE

2.3 Process Monitoring

It is not reasonable to expect that a virtual enterprise process will execute blindly. WISE must provide the necessary tools to find out the status of any process in the system in order to allow users to keep track of their processes and troubleshooting them when necessary. In addition, process design is a difficult task and it is also not reasonable that the designer will reach the best possible solution at the first attempt. Rather, process design is an iterative procedure where WISE can be of great help by providing accurate measurements of all the characteristics affecting the execution of a process: overall duration, bottlenecks, relative duration of each task with respect to the duration of the entire process, loads at each participant site, deadlines missed, and so forth.

In order to provide all this functionality, WISE incorporates the necessary modules within the execution engine to keep track of executing processes. In addition, it uses a history space where information about all already executed processes is stored and organized in a way that facilitates its analysis. We refer to this part of WISE as the process data mining tool since it will act as a repository where designers can explore existing data to find out relevant information about process execution. For displaying this information, we plan to take advantage of the sophisticated capabilities of Structware which already provides graphical user interfaces to display process execution information. In the same way that a Structware process is compiled and translated into notation understandable by the WISE engine, the information produced by the WISE engine will be translated into the appropriate format to be displayed using Stuctware's interface.

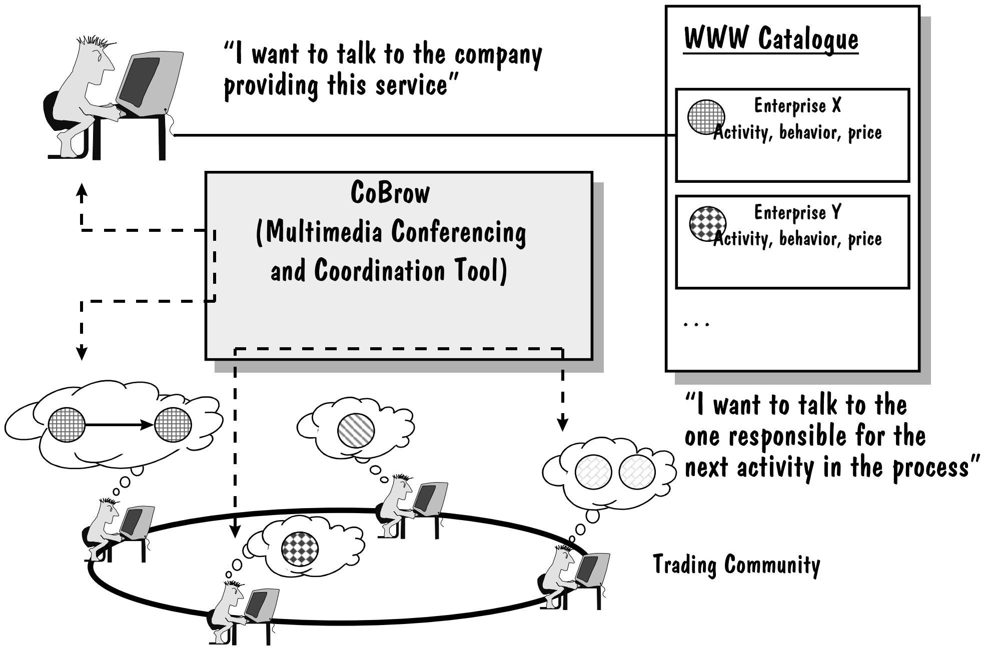

Figure 6: Coordination and communications among the elements of WISE

2.4 Coordination

Unlike in conventional workflow engines, WISE will operate in an environment where the different participants and the different elements of the process are not necessarily in a position to easily exchange information among them. Note that as the concept of virtual enterprise implies, each participant could be located not only on a different location but in an entirely different company. It is nevertheless important for the participants to be able to communicate in order to resolve the unavoidable inconsistencies and minor problems associated with any process (Figure 6). An essential aspect of this communication and collaboration is that it will be context based. That is, a user will not necessarily ask to communicate with a concrete person but, rather, with the person who played a given role in the execution of the process. If, for instance, there is a problem with some of the process data (wrong insurance number, wrong address, invalid code, etc.), a participant may request to send a message or to contact the person who introduced the data. It is in this sense that the communication and coordination will be context based and, for the same reasons, that an awareness model is necessary.

To achieve this goal, we plan to use some of the results of the CoBrow (Collaborative Browsing in Information Resources) project [SSW97] to avoid having to develop the communication infrastructure as part of WISE. Within WISE, we will focus on the awareness model and how we can take advantage of the communication facilities provided by CoBrow to implement the context based coordination environment of WISE.

3 Project Status

The first prototype of WISE was completed in April 1998. It includes the enactment functionality and a primitive form of monitoring, along with interfaces to a variety of key applications such as other workflow engines, SAP/R3, and commercial databases (Oracle, ObjectStore). The prototype has started to attract considerable attention from researchers in the areas of electronic commerce and workflow management since it show the feasibility of the ideas and provides an innovative solution to common workflow problems.

In terms of the functionality incorporated in the first prototype, we have taken advantage of several ongoing research efforts. In particular, the prototype combines the OPERA process support engine [AHST97] (developed by the Information and Communication Systems Research Group) with StructWare, the commercially available process modelling tool of IvyTeam. In addition, the prototype also incorporates capabilities to interoperate with several commercial tools such as IBM's FlowMark, SAP-R/3, and Oracle's database management system, which has been done in part by incorporating results of past projects in the area of coordination of CIM systems [HST98].

The current version of the prototype uses a simplified version of the OPERA engine to drive the execution of meta-processes in which individual tasks are processes defined in commercial workflow engines residing in different systems (FlowMark and SAP-R/3). The process definition is done through StructWare, whose output is processed and converted into OPERA's internal representation for execution. Execution of the processes can be monitored and controlled via a graphical interface showing the progress of the execution and allowing to stop, resume, and abort it at any time. The prototype can also cope with platform heterogeneity: it transparently combines components residing in UNIX, OS/2, and Windows-95 computers, both its own components (Structware runs under Windows-95, OPERA under UNIX, and the application agents in UNIX, OS/2, and Windows-95) and invoked applications (which can run under any operating system for which an application agent exists). This is a crucial point as this is likely to be the typical heterogeneous environment of a virtual enterprise.

In the future, we will also incorporate results from other research projects such as CoBrow [SSW97] (for the coordination aspects) and IvyBeans [Lie98] (for the catalogue),

4 Conclusions

WISE should be seen as an integration effort where several known technologies as well as new ideas are being brought together in order to provide a coherent technological solution. We believe that, within WISE, not only the system and overall approach is novel but also that many of the technology being developed to implement important functionality of WISE is also quite innovative. We expect that the results of the project will both enhance considerably the scope of application and expressive power of current workflow systems and open up significant opportunities in the area of electronic commerce. The WISE platform will also constitute a good testbed in which to develop new ideas and systems for electronic commerce without having to devote a considerable effort to side issues such as execution, distribution, or definition of processes.

Project Data

The WISE project is funded by the Swiss National Science Foundation. It started in December of 1997 and will have a duration of 29 months. There are three academic and two industrial partners in the project. On the academic side, the participants are the Database Research Group[1], the Computer Engineering and Networks Laboratory[2], and the Information and Communications Systems Research Group[3] of ETH Zurich. The industrial partners are IvyTeam[4], and *online*SOLUTIONS[5].

Acknowledgments

We would like to thank all those students who have helped or are currently helping in the development of WISE: W. Bausch, C. Cañamero, S. Middendorf, F. Nussberger, C. Schuler, D. Seitz, O. Separovic, and M. Vinzens.

References

[AHST97] G. Alonso, C. Hagen, H.J. Schek, and M. Tresch. Distributed Processing over Stand-alone Systems and Applications. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB'97)*, Athens, Greece, August 1997.

[HA98a] C. Hagen and G. Alonso. Beyond the black box: Event-based inter-process communication in process support systems. Technical Report 303, ETH Zurich, Institute of Information Systems, July 1998.

[HA98b] C. Hagen and G. Alonso. Flexible exception handling in the OPERA process support system. In *Proc. of the 18th Intl. Conference on Distributed Computing Systems*, Amsterdam, The Netherlands, May 1998.

[Hag97] C. Hagen. Atomarität in Workflow- und Prozessunterstützungssystemen. In *9. GI-Workshop "Grundlagen von Datenbanken"*, Friedrichsbrunn, Germany, May 1997. In German.

[Lie98] H. Lienhard. IvyBeans - Bridge to VSH and the project WISE. In *Proceedings of the Conference of the Swiss priority Programme Information and Communication Structures, Zurich, Switzerland*, July 1998.

[SAS98] H. Schuldt, G. Alonso, and H.-J. Schek. A Unified Theory of Concurrency Control and Recovery for Transactional Processes. Technical report, ETH Zurich, 1998. In preparation.

[SST98] H. Schuldt, H.-J. Schek, and M. Tresch. Coordination in CIM: Bringing Database Functionality to Application Systems. In *Proceedings of the 5th European Concurrent Engineering Conference (ECEC'98), Erlangen, Germany*, April 1998.

[SSW97] G. Sidler, A. Scott, and H. Wolf. Collaborative Browsing in the World Wide Web. In *Proceedings of the 8th Joint European Networking Conference, Edinburgh, Scotland*, May 1997.

[1] http://www-dbs.ethz.ch/

[2] http://www.tik.ee.ethz.ch/

[3] http://www.inf.ethz.ch/department/IS/iks/

[4] http://www.ivyteam.com/

[5] http://www.online-solutions.ch