# 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

## Abstract

*The Internet and the proliferation of inexpensive computing power in the form of clusters of workstations or PCs provide the basic hardware infrastructure for business to business electronic commerce in small and medium enterprises. Unfortunately, the corresponding software infrastructure is still missing. In this paper we show a way to develop appropriate tools for electronic commerce by describing the approach we have taken in the WISE project (Workflow based Interned SErvices). The goals of WISE are 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.*

## 1 Introduction

Historically, companies have found many ways to work together in regard to suppliers, partnerships, customers, etc. Typically, these ventures have taken place without computer support: members of each company got together in a shared space to work on a particular project. Only a few large companies could afford the investment required to map these commercial ties to an electronic infrastructure, which, until recently, was usually based on expensive leased lines, mainframes and ad hoc development. Even for these companies, the expense could only be justified above certain levels of trading, with partners who could also afford the infrastructure, and with whom cooperation would continue on a long term basis [1].

Many of the barriers to access such technology have been effectively eliminated by the Internet [2, 3], which offers a relatively inexpensive communication medium and allows to initiate business-to-business ventures in a much more dynamic manner. Nowadays, the remaining obstacle is the lack of appropriate software infrastructure.

To address this limitation, the WISE 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. 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 catalogue 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, catalogue based information, catalogue search, and communication frameworks. In this extended abstract, we briefly describe the goals and most relevant characteristics of the project. Due to the lack of space, we limit ourselves to provide an overview of what we believe to be the most innovative ideas in the project. We see these ideas as crucial steps in any attempt to provide software support for e-commerce and, thus, the paper's contribution is to spell out all the key issues that need to be addressed in order to provide a complete solution.

The paper is structured as follows. Section 2 presents an example of the type of virtual enterprise to be supported by WISE and the technical challenges to be faced when implementing such functionality. Sections 3 to 6 describe the main components of the WISE system. Section 7 presents the current status of the development efforts and section 8 concludes the paper.

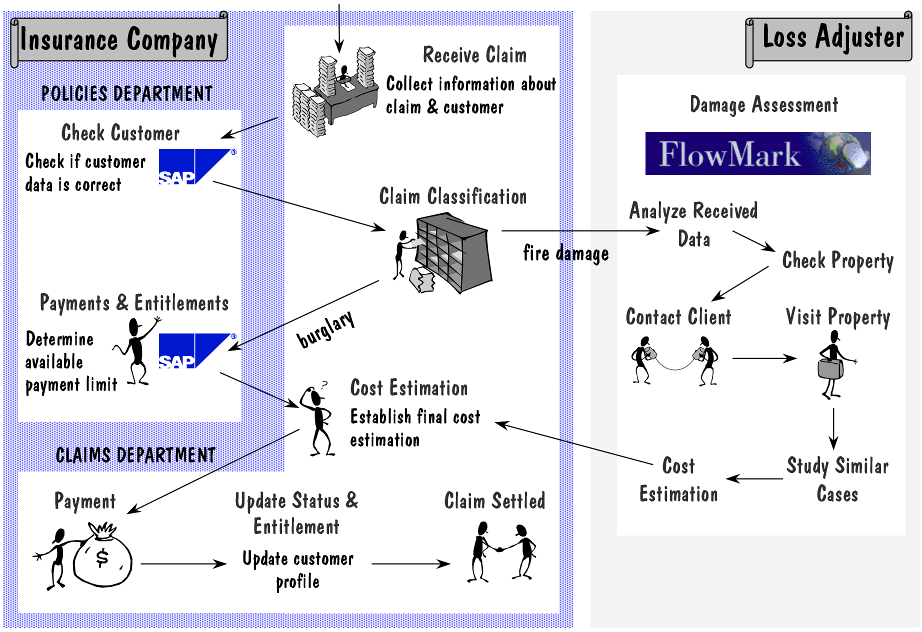

**Figure 1. Example of a virtual enterprise process**

## 2 Motivation

In this section we first present a typical example of virtual enterprise process to motivate the functionality we have implemented and then discuss the technical challenges.

### 2.1 Virtual Enterprises

The WISE approach is based on the notions of trading community, virtual enterprise, and virtual business process. A virtual business process uses information and communication technology to allow a business process [18] to go beyond the corporation boundaries. A virtual enterprise is thus defined as an enterprise whose business processes are virtual business processes. These virtual business processes are constructed by combining the services provided by different companies. The set of companies working together as part of a virtual enterprise is what we refer to as a trading community.

As an example consider the scenario shown in Figure 1 (which has been implemented as part of the WISE project). In this scenario, the trading community is formed by two different departments of an insurance company and a loss adjuster company. The insurance company defines a virtual business process to handle insurance claims. In the first step of the process, a clerk in the "claims department" receives the claim and collects all the necessary information about the claim itself, the customer, the involved parties, etc. This information is processed in the "policies department", which uses SAP as the supporting tool. In this department, the data provided is correlated with the information available in the database, i.e., whether the customer is up to date in payments, whether it is a case covered by the existing policies, and so forth. Once this step is completed, the information is forwarded back to the claims department where the claim is classified, i.e., the specific type of claim (burglary, flood, fire, car accident, damages by third parties, etc.) is determined.

For the purposes of this example, we will consider only two types of claims: burglary and fire. In case of burglary, the claim is returned to the policies department where, based on the police report, the total value of the objects is calculated, the payment limit established and an estimate is made of how much the insurance should pay. In case of fire damage, the process is more complicated. To deal with such cases, the insurance company resorts to a loss adjuster company which will be the one responsible for making an estimate of what needs to be paid. After the estimation is completed, the payment is made, the corresponding records updated (so that a customer is not paid several times for the same claim), and the claim settled.

The interesting part of this example is the introduction of the loss adjuster process as one more element of the overall claim processing procedures, even if the loss adjuster is an entirely different company. The key to the WISE approach to virtual enterprises is to treat such interoperability problems as process encapsulation problems where, as long as there are well defined input and output parameters, the rest can be treated as a black box.

In the example, the loss adjuster, who uses a workflow engine to drive its business processes, provides an entry point similar to an API which the insurance company can invoke. Through this interface, the loss adjuster receives the necessary data and triggers its own business process. This process consists of checking the property, i.e., who is the legal owner of a building, arrange a meeting with the client, visit the damaged property, compare with similar cases or, in case of major disasters like floods or earthquakes, determine what other sources of payment may need to be considered, and a cost estimate is made. The cost estimate is then forwarded to the insurance company, which using a similar mechanism can incorporate this step into its own business process.

## 2.2 The technical challenge

The example above serves to illustrate the difficulties that need to be solved in order to implement our notion of virtual enterprise.

First of all, it must be pointed out hat this is not just an interoperability problem. In fact, our experience shows that interoperability is easier to address once the notion of virtual process is in place, since it provides a much narrower and well defined framework in which to develop a solution. This is proven by the fact that we have had no problem in linking different applications residing on different operating systems once the virtual business process had been defined. We believe the real challenge in electronic commerce is how to provide a complete solution. In our case, this means to develop a software tool capable of supporting the entire life cycle of a virtual business process. We see these business processes as valuable assets which need to be not only defined and enacted but also maintained, updated, and monitored. Otherwise, the solution provided suffers from the same drawbacks as current practices: development is ad-hoc and costly, almost impossible to maintain, and limits the applicability of electronic commerce to a very few special cases. Moreover, we believe that what is needed is a generic framework which can be used to develop virtual business processes without a significant amount of expertise or development cost. This framework should provide solutions to non-trivial problems such as how a company incorporates the services of another as part of its own business processes, how a company can advertise its own services and make them available to other companies, or how a virtual business process can be enacted and its execution monitored, just to mention a few. Without these solutions, the notions of virtual enterprise, trading community, and virtual business process described above may be conceptually appealing but become irrelevant for practical purposes.

## 2.3 Complete Solution

The WISE project is an integration effort with the final goal of providing a complete solution. In order to do this, there are four issues that need to be addressed:

- How to provide a mechanism whereby companies can advertise their services, other companies can look at them and, finally, incorporate these services into their own business process without requiring ad-hoc development.
- How to execute processes spawning several companies and which, ultimately, may be entirely virtual in the sense that no physical company is responsible for driving the main process.
- How to keep track of such processes and provide the same monitoring and analysis tools that exist for normal business processes.
- How to establish a framework for context specific collaboration, i.e., one in which communication is not based on point-to-point routing but based on the dynamics of the process execution.

The WISE architecture (Figure 2) is organized into four components (definition, enactment, monitoring, and coordination), each one of them with the role of addressing a particular issue of the four listed above. Thus, the *process definition* component allows virtual business processes to be defined using as building blocks the entries of a catalogue where companies within a trading community can post their services. Similarly, the *process enactment* component compiles the description of the virtual business process into a representation suitable for enactment and controls the execution of the process by invoking the corresponding services of the trading community. The *process monitoring and analysis* component is a tool keeping track of the progress made in the execution of the virtual business process and of the status of all active components in the system. The information produced by these tool is used to create an awareness model [10] used for load balancing, routing, and quality of service purposes as well as, later on, for analysis of the behavior of the process. Finally, the *coordination and communication* component supports multimedia conferencing and cooperative browsing of relevant information between all participants in the trading community using the information produced by the business process as the main source for routing. It is in this sense that the communication in the trading community is context based since it is established on semantic information derived from the execution of the virtual business process.

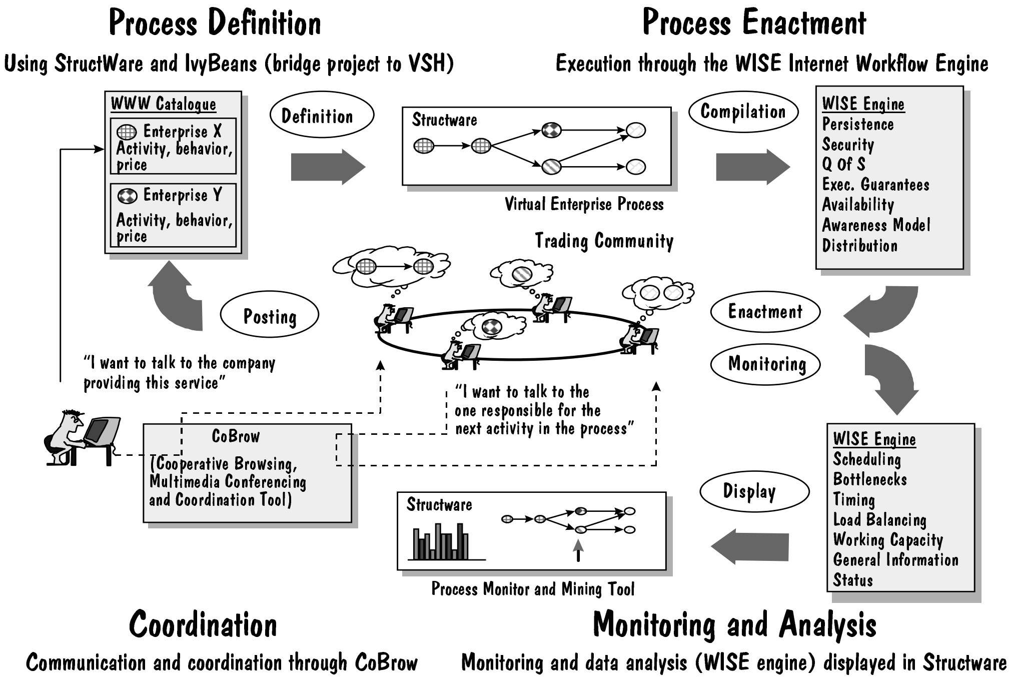
Process Definition
Using StructWare and IvyBeans (bridge project to VSH)
WWW Catalogue
Enterprise X
Activity, behavior, price
Enterprise Y
Activity, behavior, price
Definition
Structware
Virtual Enterprise Process
Process Enactment
Execution through the WISE Internet Workflow Engine
Compilation
WISE Engine
Persistence
Security
Q Of S
Exec. Guarantees
Availability
Awareness Model
Distribution
Trading Community
Posting
Enactment
Monitoring
"I want to talk to the company providing this service"
"I want to talk to the one responsible for the next activity in the process"
CoBrow
(Cooperative Browsing, Multimedia Conferencing and Coordination Tool)
Structware
Process Monitor and Mining Tool
Display
WISE Engine
Scheduling
Bottlenecks
Timing
Load Balancing
Working Capacity
General Information
Status
Coordination
Communication and coordination through CoBrow
Monitoring and Analysis
Monitoring and data analysis (WISE engine) displayed in Structware

**Figure 2. The different components of the WISE project**

In WISE, these four components are tightly integrated reflecting an approach to electronic commerce based on transparency and ease of use: the functionality of each component must always be considered in relation to that of all other components. While there is innovation in each individual component, our measure of success is the degree of integration of the system as a whole.

## 3 Process definition

In WISE, virtual business processes are constructed by using the services offered by different companies as building blocks. 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 business 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 publish their services. The second is a way to define a process based on such services. For these purposes, WISE uses a WWW catalogue and a business process modeling tool (Figure 3).

The WWW catalogue uses Java applet/servlet technology to allow companies in the trading community to advertise their services and to "see" the semantics of the services provided by other companies [19]. While a list of mere "read-only" pointers would seem to suffice, in practice companies need to understand the behavior of a service before they can incorporate it into their business process. The catalogue, instead of just URL's, contains objects encapsulating the behavior of a each service. A Java version of a business modeling tool supporting simulation and analysis (see below) is then used to allow a company to see the exact characteristics of each entry in the catalogue, not only in terms of the steps it takes but also in terms of its functional specification: cost, average duration, guarantees, requirements, side effects, etc. When a company wants to make an entry in the catalogue, it specifies the service using the modeling tool. This generates code that is inserted in the catalogue and executed in the simulation and analysis tool every time another company is interested in using the service as a step within a virtual business process. Note that the specification of the service does not represent additional overhead. Such specifications are necessary independently of the way the virtual enterprise is implemented.

From the catalogue, a drag and drop type of interface is used to build the virtual business process. The tool we

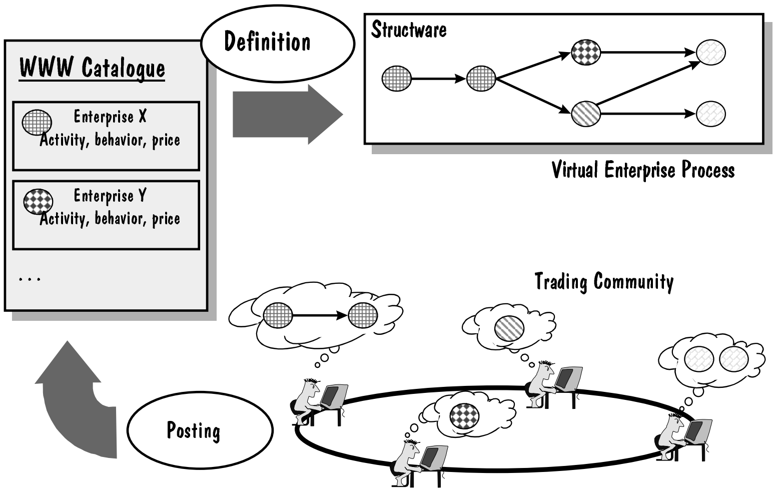

**Figure 3. Process definition within WISE**

use for process definition is *Structware* [15], a product of IvyTeam, one of the partners in the project. Structware, which is internally based on Petri-nets, supports not only the modeling of business processes but also sophisticated analysis of its behavior (bottlenecks, average execution times, costs, delays, what if analysis, etc.). This analysis capability is the one used in the catalogue, but it can also be used to analyze the behavior of the virtual business process once it is constructed.

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 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 tool, we can incorporate as well the services posted to the catalogue without having to worry at this stage about how they will be invoked when the process needs to be executed. The necessary invocation parameters are provided by the company posting the service. 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 not available in commercial workflow tools. For instance, we can provide a complete exception handling capability [14] with characteristics 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 [13]. This functionality is missing in current systems and we consider it to be crucial in realistic environments.

## 4 Process enactment

The enactment of the virtual business processes is performed by the WISE engine, which is based on work done within the OPERA project [12, 5]. The WISE engine extends ideas from workflow management [9, 6], and uses known techniques for distributing this functionality [23, 16, 7]. In addition, a considerable amount of extensions have been introduced to make workflow a suitable foundation for electronic commerce (for a different approach to electronic commerce based on workflow technology see [20]). Among them, there are three that deserve special attention: security, quality of service, and execution guarantees.

Given the nature of the data exchanged between the different participants in the trading community, WISE incorporates 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 incorporates quality of service guarantees based on execution statistics and network characteristics. Our current approach is based on distinguishing different process categories. Each one of these categories provides a different quality of service. At the top, *critical* processes cause the system to use a RSVP protocol (resource reservation protocol) to guarantee the bandwidth necessary to maintain a given throughput and response time. *Important* processes, on the other hand, cause the engine to use different techniques to guarantee certain maximum delays in the execution. While these guarantees are not as strong as for critical processes, they still allow to bound the time a process will be delayed. Finally, *normal* processes are executed in a best effort mode, that is, without guarantees regarding possible delays. The necessary information for reserving services and providing guarantees is derived from the awareness model created by the monitoring and analysis component.

Finally, the WISE engine also incorporates execution guarantees, whereby a process is always 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 [21]. The execution guarantees are based on the notion of spheres of atomicity and isolation, which allow us to specify which parts of the business process need to be made atomic for recovery purposes and which parts of the process need to be isolated from interferences of other processes. The notion of spheres has proven to be very useful to formally formulate the properties a process needs to meet in order to guarantee correctness from a transactional point of view [11, 8, 4, 17].

## 5 Process Monitoring

It is not reasonable to expect a virtual enterprise process to execute blindly. Accordingly, WISE provides tools to find out the status of any process in the system in order to allow users to keep track and troubleshoot them when necessary. In addition, process design is a difficult task. It is also not reasonable to expect the designer to reach the best possible solution at the first attempt. In particular, in virtual enterprise environments it is difficult to foresee all possible eventualities until some example runs are available. 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 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. For displaying this information, we plan to take advantage of the 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.

Finally, WISE will also include an awareness model [10] 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.

## 6 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 trading community implies, each participant could be 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 4). 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.

To achieve this goal, WISE uses the results of the CoBrow (Collaborative Browsing in Information Resources) project [24] to avoid having to develop the communication infrastructure as part of WISE.

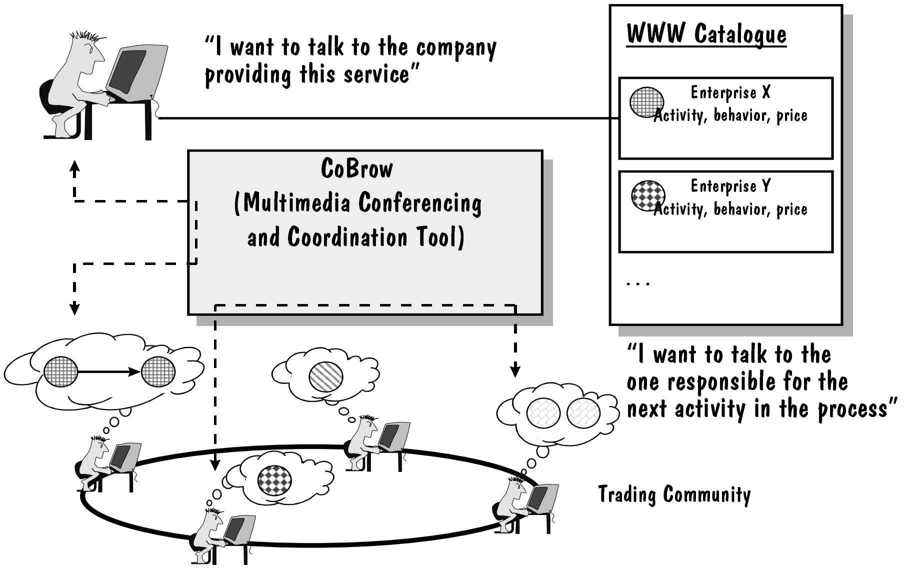
"I want to talk to the company providing this service"
WWW Catalogue
Enterprise X
Activity, behavior, price
Enterprise Y
Activity, behavior, price
...
CoBrow
(Multimedia Conferencing and Coordination Tool)
"I want to talk to the one responsible for the next activity in the process"
Trading Community

**Figure 4. Coordination and communications among the elements of WISE**

## 7 The WISE system

The first prototype of WISE 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 shows the feasibility of the ideas and provides an innovative solution to common workflow problems. The first prototype is based on several ongoing research efforts. In particular, the prototype combines the OPERA process support engine [5] with StructWare, the commercially available process modeling tool of IvyTeam. In addition, the prototype also incorporates capabilities to inter-operate 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 [22].

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 systems residing in different systems (FlowMark and SAP-R/3) connected via Internet. 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 [24] (for the coordination aspects) and IvyBeans [19] (for the catalogue).

## 8 Conclusions

In this paper, we have presented a basic infrastructure for business to business electronic commerce. In this form of e-commerce, different companies join their services to form a virtual enterprise, which provides a business process that can be executed over the Internet. WISE includes different components to define, enact and monitor visual enterprise processes, supporting also the communication and coordination between the participants.

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, the Computer Engineering and Networks Laboratory, and the Information and Communications Systems Research Group of ETH Zürich. The industrial partners are IvyTeam, and *online*SOLUTIONS.

## 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. We would like to thank our industrial partners in the project, IvyTeam and *online*SOLUTIONS, for their support and valuable input to the ideas presented here.

## References

[1] Survey: Electronic commerce. *The Economist*, September 1997. http://www.economist.com /editorial /freeforall /14-9-97 /index_survey.html.

[2] David J. Abel. Spatial internet marketplaces: A grand challenge? In *Proceedings of the 5th International Symposium on Spatial Databases (SSD'97), LNCS*, pages 3–8, Berlin, Germany, July 1997. Springer-Verlag.

[3] David J. Abel, Volker Gaede, Kerry L. Taylor, and Xiaofang Zhou. SMART: Towards Spatial Internet Marketplaces. Technical Report 97/28, CSIRO Mathematical and Information Sciences, GPO Box 664, Canberra, ACT 2601, Australia, 1997.

[4] G. Alonso, D. Agrawal, and A. El Abbadi. Process Synchronization in Workflow Management Systems. In *8th IEEE Symposium on Parallel and Distributed Processing (SPDS'96). New Orleans, USA.*, October 1996.

[5] 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.

[6] F. Casati, P. Grefen, B. Pernici, G. Pozzi, and G. Sanchez. WIDE Workflow Model and Architecture. Technical Report 96-19, University of Twente, 1996.

[7] Stefano Ceri, Paul W.P.J. Grefen, and Gabriel Sanchez. WIDE: A Distributed Architecture for Workflow Management. In *Proceedings 7th International Workshop on Research Issues in Data Engineering (RIDE'97)*, pages 76–79, Birmingham, UK, April 1997.

[8] D. Georgakopoulos and M. Hornick. A Framework for Enforceable Specification of Extended Transaction Models and Transactional Workflows. *International Journal of Intelligent and Cooperative Information Systems*, 3(3), September 1994.

[9] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, April 1995.

[10] Dimitrios Georgakopoulos. Collaboration management infrastructure for comprehensive process and service management, May 1998. Presentation in International Symposium on Advanced Database Support for Workflow Management, Enschede, The Netherlands.

[11] Dimitrios Georgakopoulos, Mark Hornick, Piotr Krychniak, and F. Manola. Specification and Management of Extended Transactions in DOMS. In *RIDE-IMS'93*, pages 253–257, Vienna, Austria, April 1993.

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

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

[14] 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.

[15] IvyTeam. Structware'98 Process Manager. Available through http://www.ivyteam.com, 1998.

[16] S. Jablonski and C. Bussler. *Workflow Management*. International Thomson Computer Press, 1996.

[17] D. Worah and A. Sheth. Transactions in Transactional Workflows. In *Advanced Transaction Models and Architectures*, Sushil Jajodia and Larry Kerschberg, editors., chapter 1, pages 3–34. Kluwer Academic Publishers, 1997.

[18] F. Leymann and W. Altenhuber. Managing Business Processes as an Information Resource. *IBM Systems Journal*, 33(2):326–348, 1994.

[19] 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.

[20] P. Muth, J. Weissenfels, and G. Weikum. What Workflow Technology Can Do For Electronic Commerce. Technical report, University of the Saarland, Department of Computer Science, Saarbrücken, Germany.

[21] H. Schuldt, G. Alonso, and H.-J. Schek. A Unified Theory of Concurrency Control and Recovery for Transactional Processes. Technical report, ETH Zurich, 1998. Available at: http://www-dbs.ethz.ch/˜schuldt/Publications/sas98.ps.

[22] 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.

[23] H. Schuster, S. Jablonski, T. Kirsche, and C. Bussler. A Client/Server Architecture for Distributed Workflow Management Systems. In *Proc. of Third Int'l. Conf. on Parallel and Distributed Information Systems*, Austin, Texas, September 1994.

[24] 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.