

Towards a Software Architecture for Distributed and Mobile Collaborative Systems

Schahram Dustdar and Harald Gall

*Distributed Systems Group, Technical University of Vienna, Austria
{dustdar, gall}@infosys.tuwien.ac.at*

Abstract

Current trends in collaborative business emphasize the importance of business process support within and between organizations. These process participants are increasingly distributed and mobile. It remains a challenge to utilize software as to share and exchange ideas and work on collaborative activities across locations and different mobile devices, while still business process aware. In this paper we develop a five-layer software architecture for distributed and mobile collaborative (DMC) systems, which provides mobility of context to its group members. This architecture defines a foundation for the flexible integration of Collaborative Systems (such as Workflow Management, Groupware or Business Process Modeling) with teamwork services that support distributed and mobile collaboration.

1. Introduction

In the course of the past decade, there has been a lot of work in developing middleware for integrating, automating, and augmenting business processes. Current trends in collaborative business emphasize the importance of business process support inside organizations as well as between enterprises. One of the fundamental problems is the lack of business process support for distributed and mobile collaborative (DMC) systems. Software systems such as Workflow Management Systems (WfMS) and Groupware systems have been used to automate or to augment business processes in organizations [5, 7, 9, 10, 25]. WfMS have been defined as “technology based systems that define, manage, and execute workflow processes through the execution of software whose order of execution is driven by a computer representation of the workflow process logic” [26]. Current WfMS focus on automating structured (modeled) intra-organizational business processes. Groupware typically does not contain any knowledge or representation of the *goals* or underlying business *processes* of the group [11, 13, 15, 27]. On the other hand, WfMS are typically “organizationally aware” because they contain an explicit representation of organizational processes (process model). However traditional WfMS present a rigid work environment consisting of *roles* and their associated

activities and applications. In this context they do not provide support for frequently changing process participants, ad-hoc formation of groups collaborating on a business process, and device independent support of group activities.

In recent years there has been considerable attempts to merge workflow-, groupware-, and business process modeling technologies [8]. Industrial research labs and product teams have made significant steps forward [3, 16]. Future WfMS have to cover inter-organizational activities and processes including product value-chains on the Internet [18, 21, 24, 28] regardless of location (mobility) and regardless of devices used. A WfMS can impose a rigid work environment on users, which often has implications: One example is among users who perform time-consuming manual “work arounds”; the consequence is lower efficiency and overall dissatisfaction with the system. As a result WfMS enable continuous loops of sub processes such as goal setting, working, monitoring the work, measuring performance, recording and analyzing the outputs and evaluating the “productivity” of personnel. Users of WfMS often consider the controlling and monitoring possibilities as a “dark side” of these systems, which results in demotivating employees and a lack of support for distributed and mobile work.

Current WfMS and Groupware systems do not combine features regarding *flexibility*, *adaptability*, and *traceability* of process support for collaborative work environments. Future distributed and mobile collaborative systems need to facilitate not just mobility of content to group members, but also *mobility of context* of activities in business processes, i.e. providing information about process instances, their associated artifacts, and connectivity modes of group members (such as fixed, mobile, or ad-hoc).

The contribution of this paper is as follows: Firstly, it presents an overview of current trends and categorization of discussed collaborative technologies used in organizations. Secondly, it presents challenges and unresolved issues in the context of distributed and mobile collaboration. Thirdly, it proposes a software architecture for distributed and mobile collaborative systems, which aims to overcome current limitations of WfMS and Groupware systems, namely to provide required *mobility*

of context to group members. The described architecture is based on research results and a running prototype system of the EU framework V project MOTION (Mobile Teamwork Infrastructure for Organizations Networks).

2. Distributed and mobile collaborative systems grid

Recent advances in the area of Internet Computing and collaborative business process management systems (WfMS, Business process modeling systems, Groupware) are often seen as key enablers for facilitating business processes in organizations. Cooperative tasks in teams are increasing, and as a consequence the use of collaborative systems is becoming more pervasive. To understand current collaborative technologies it is important to first analyze current systems. In this paper we present a “distributed and mobile collaboration grid”, which distinguishes two dimensions: *Process* and *Connectivity* as shown in Figure 1.

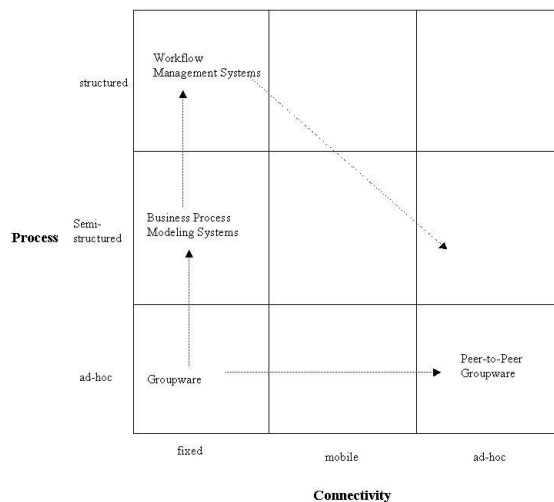


Figure 1. Distributed and mobile collaboration grid

A business process can be ad-hoc, semi-structured, or structured (modeled). For example a business process such as “customer order entry” can be modeled using a traditional WfMS. However a *structured* process can only be enacted (instantiated) as it was designed. If an exception occurs, a workflow administrator needs to re-model the process before the execution can continue. This limits the usability of workflow systems in a world where constant adaptation to new situations is necessary and where teams are increasingly mobile and distributed. An example of an *ad-hoc* process is the discussion of a project’s design review using Groupware. A *semi-structured* process consists of groups of activities, which are modeled; however in contrast to a structured (modeled) process it consists also of activities, which are not pre-defined.

The second dimension presented in the DMC grid is *Connectivity*. Here we distinguish between three modes: fixed, mobile, or ad-hoc. We speak of *fixed* connectivity when users work on computers permanently connected to a network. For example in an office where each employee has a personal computer connected to the company-wide network or a wide area network (WAN). *Mobile* connectivity essentially describes a mode where people are “on the move” but access data and applications located on their remote network. The *ad-hoc* mode allows users to establish a “virtual” group of users on the fly. Participants in ad-hoc groups may have network connectivity either permanently or sporadically.

The mobility of participants also offers new ways of distributed collaboration: processes are no longer bound to locations of resources (such as participants or artifacts) but can consider several availability modes.

Technologies used in organizations today basically can be associated to one cell of the grid in Figure 1. The arrows in Figure 1 depict the evolution of recent technology developments. Workflow management systems traditionally provide support for structured processes (y-axis) and fixed connectivity (x-axis). WfMS are being enhanced to also provide support for mobile and ad-hoc modes of connectivity. This trend increases the flexibility, adaptability and traceability of process activities in WfMS to support mobility of context for business processes. Business Process modeling systems allow modeling of semi-structured and structured processes (y-axis) but require fixed connectivity of users to a network. Groupware systems provide support for ad-hoc processes (synchronous or asynchronous). In most cases groupware requires fixed connectivity. Technologies such as Peer-to-Peer or mobile device support for Java facilitate the trend towards Peer-to-Peer Groupware, which per definition enables users to establish ad-hoc structures (y-axis) and in the same time ad-hoc connectivity (x-axis).

To summarize the challenges: To build software systems supporting fully distributed and mobile collaboration requires functionalities currently found in *different* software application domains such as WfMS, Groupware, or Business Process modeling tools. A challenge for future DMC systems is to develop a sustainable software architecture, which provides support for both dimensions presented in Figure 1 and is flexible, adaptable, and traceable regarding processes of collaborative work. It is of paramount importance for a DMC system to enable geographically dispersed users with different modes of connectivity (fixed, mobile or ad-hoc) to share information in various kinds such as using middleware and/or peer-to-peer technologies. Users need to register themselves and receive notifications on events, regardless of their location or device they use. Collaboration partners need to be empowered to locate each other, find experts in required domains and link all coordination information with artifacts such as documents. Therefore the mobility

of context (who, what, why, when and using which resources) is essential for DMC systems.

Example *use cases* of distributed and mobile collaboration include: information updating and notification of availability (of resources); searching and inviting people for diverse synchronous communication (e.g. chat, video/telephone conference); expert search; information retrieval about resources (e.g. users, artifacts, processes); synchronous and asynchronous communication in a community; synchronous collaboration on artifacts (e.g. Groupware); community establishment and updating.

3. Distributed and mobile collaborative systems architecture

Based on the recent results in software architecture research and practice [2, 4, 17, 23] we adopt the quasi-standard terminology to describe a DMC architecture: A software architecture typically includes the description of *components*, *connectors*, and *configurations* [23]. In terms of DMC systems these terms have to be discussed from a mobile and distributed collaboration perspective.

Since such an architecture has to cope with fixed, mobile and ad-hoc connectivity modes we decided to strive for a peer-to-peer (P2P) style rather than a classical client-server (CS) style. P2P facilitates ad-hoc meetings and information sharing without the presence of some particular server; but it also offers ways to exploit CS structures in supporting distributed and mobile collaboration (e.g. persisting artifacts, distributing information using hierarchies of computers etc.).

Our DMC architecture has a P2P nature in cases where this is beneficial but also exploits classical CS structures where appropriate. The following descriptions will point out the respective architectural style used in a particular layer or component.

3.1 Conceptual Architecture View

Before describing each of the components depicted in Figure 2, we group them into logical units with clear responsibilities, following the principles of Parnas [22]. A DMC system consists of five layers:

- The *Communication Layer* provides communication means between peers and their software components; it is a middleware layer that supports P2P protocols such as Gnutella or P2P architectures such as JXTA [14] or Gridella [1].
- The *Basic Services Layer* contains responsibilities for artifact and user (participants) management, process configuration and composition, publish and subscribe, distributed searches on all kinds of resources, messaging, and notification.
- The *Teamwork Services Layer* provides a uniform access point for all kinds of teamwork services that can be used by higher-level applications such as

WfMS or Groupware in a DMC context. This includes a common teamwork services API as well as DMC based service presentation.

- The *Service configuration layer* offers means to configure services of the Teamwork Services Layer according to application-specific requirements of WfMS or Groupware systems. Further, this configuration also includes user-specific configurations such as reachability data (e.g. e-mail, phone, SMS etc.)
- The *Presentation Layer* shields the applications from device-dependent GUI adaptation and information tailoring by abstracting from particular end-user devices and their constraints (i.e. display sizes, computing and platform constraints etc.).

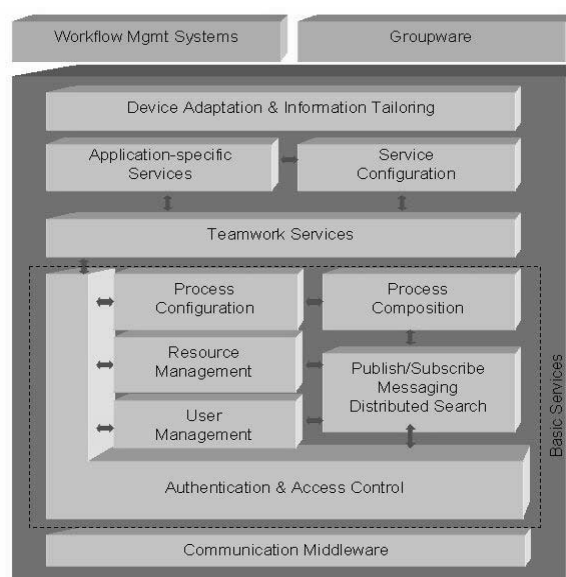


Figure 2. DMC Conceptual Architecture

3.2 DMC components

In this section we describe the most important components of the DMC architecture in terms of their responsibilities and abstractions. We thereby focus on the connectivity and process axes of the DMC grid as basis for WfMS and Groupware systems, which are denoted as Basic Services in Figure 2. Users (or process participants) should be granted access via various types of devices ranging from PCs, notebooks to PDAs or mobile phones for fixed, mobile or ad-hoc connectivity.

Participants can be addressed and reached via the concept of a *community* that resembles a team or a group of interest. This concept allows building communities for specific purposes and tasks as the basis of distributed and mobile collaboration of people. Both participants and artifacts are connected in communities and share their information in a peer-to-peer style.

User (community) management includes setup and configuration of community leaders, community members

and also community friends (as a more loosely coupled variant of a team member). Adding/removing participants to/from a community, giving participants specific access rights to resources etc. define the responsibilities of this component. It provides community as central abstraction to other components for addressing groups of people and sharing and exchanging information with them.

Resource management: resources include various kinds of artifacts required for a particular process (or process template) and can be of any MIME-type (text, audio, video, graphics etc.). Resource management also includes information about particular resources such as searches for artifacts (matching some user-defined criteria), notification about the availability of some artifact(s) etc. In this context information about a resource includes both meta-information about artifacts and the artifact itself. So searches and subscriptions/notifications can be handled on a meta-data level more easily and efficiently for large sets of users.

Process Configuration is concerned with managing the relationships between process participants and artifacts and providing this information to other components. Process participants may be human users or software agents (i.e. components). Artifacts may be documents or other resources such as database records or applications. Such a process configuration, for example, can be that user (process participant) "Smith" requires the document artifact "invoice" in a process (or process instance) named "Sales cycle".

Process Composition is concerned with managing process models including coordination and synchronization of its sub-processes and tasks. Each process model consists of a set of tasks. The degree of granularity of process tasks can vary. On a generic level a process model (template) consists of a directed graph consisting of tasks and connection constructors such as OR and AND. On an instance level a process model consists of instantiated tasks (activities) performed by process participants (human agents or software agents).

Publish/Subscribe and notification is a component that provides loosely coupled communication among components. Its focus is thereby on subscription to all kinds of resources (including artifacts, users, communities, processes, access rights etc.). A participant can use this functionality to declare interest, for example, in the state of a particular artifact (whenever it is changed or updated he should be notified). The same applies to users, communities, or processes. As a result this component allows notification of specific activities and can be used for process composition and configuration within or across communities.

Distributed searches are based on meta-data stored in so-called profiles. These profiles describe artifacts, users, processes, or communities in a concise way and represent it in XML form. A distributed search, therefore, queries XML repositories (of different content) on each peer and – if successful – returns the requested piece(s) of

information. This concept of meta-data search allows also for querying the DMC repositories for information that might be available in the near future and of which a user wants to be notified whenever it is available.

Distributed searches further can be used to search for experts in a particular problem domain and invite them upon availability and reachability to join a (virtual) meeting. This enables the exchange of expertise across communities and processes, which is especially important in mobile and distributed collaboration scenarios of large enterprises where people are on the move very often.

The *Authentication and Access Control* component consists of an access control system called DUMAS (Dynamic User Management System) [12] and a security component responsible for integrity, confidentiality and authentication. The access control system covers three responsibilities: user control, community control, and authorization.

These Basic Services components are abstracted and shielded by the Teamwork Services to provide uniform DMC access points. Based on that layer any specific DMC application such as WfMS or Groupware can configure the teamwork services according to their specific requirements and also build new application-specific service on top of the Teamwork Services layer. This allows to define and/or to configure application-specific services (only relevant for a particular WfMS or Groupware application) on top of the Teamwork Services API. A service configuration can, therefore, include the instantiation of processes (templates) and communities (that are a grouping of artifacts, users, and access rights) for specific tasks (e.g. holding a review meeting while people are on the move, or in different branches of the enterprise, and work on diverse devices).

Process coordination and synchronization are, therefore, enabled on a common teamwork services interface and participants can smoothly move from the ad-hoc connectivity mode to a mobile or fixed mode when carrying out their tasks and activities.

A more detailed discussion on the technical details of the components is beyond the scope of this paper, but can be found in [19].

3.3 Connectors

Connectors typically define the kind of communication that occurs between software components. The description of connectors is often enriched by information/data that is *required* and/or *provided* by a component to perform its functionality to the environment.

In our case we distinguish connectors depending on the connectivity mode. When disconnected, participants can work in their local workspace and follow pre-defined initiated processes. Artifacts and certain process information reside on their local device enabling them to continue working while not connected to a network. Components such as user (community) management or

resource management communicate via common service requests (e.g. method invocations in a JVM).

Once a participant connects to the system and is in *fixed* mode, he can share his work products with others in his community and can fully exploit the functionality of DMC. For this different communication protocols (i.e. connectors) between components come into operation: middleware protocols, RMI, or HTTP. The architecture utilizes existing Web technologies such as Universal Resource Locators (URLs) or secure HTTP connections through SSL. This allows widely available access to the DMC platform from various devices (ranging from Web-terminals at airports to full-fledged computers).

Especially interesting in the context of distributed and mobile collaboration is the *mobile* mode as a mixture of fixed and disconnected working. In this case the different communication scenarios alternate depending on the network availability.

The *ad-hoc mode* empowers users to quickly set up communities in situations where there is not network connectivity available or necessary. Processes can be instantiated from templates and information can be shared on a peer-to-peer basis allowing quick coordination and synchronization of tasks and easy information sharing within a community.

3.4 Configurations

Configurations of processes include all relationship information regarding process participants and artifacts they use during those processes. By utilizing this information, it is possible for DMC based systems to combine features regarding flexibility, adaptability, and traceability of processes. For example in a DMC based system it is possible to support collaborative work in a flexible way since the system is “aware” of relationships between artifacts used by a participant. The system is adaptable because it supports a set of modes regarding connectivity (fixed, mobile, and ad-hoc). Traceability is an important aspect for mobility of context, one of the design goals of DMC based systems. For distributed and mobile collaborative work it is essential to provide process state and artifact information in a location independent manner.

3.5 Architectural Principles of DMC

One of the advantages of our DMC architecture is that it foresees possibilities for the alternation of connectivity modes:

- Peer-to-peer data sharing: data within a community are shared on a peer-to-peer basis, which means that resources of one peer are only available to others while the peer is connected.
- Server-based data sharing: artifacts that need to be available all the time – despite the connectivity of peers – can be placed on peers that belong to the

backbone of the system and are always up and running.

- Server- and Peer-based meta-data sharing: profile information about different resources (users, communities, artifacts etc.) is shared (and can be searched for) among peers. Peers of the backbone can be used to host profile information allowing distributed searches despite actual peer connectivity. Searches result in meta-data information about, for example, experts, artifacts, process states, etc. without exposing the actual information source in detail (results include the URL of a document for later retrieval by the interested participant).

Further advantages stem from collaboration aspects: Process participants may collaborate independently from their location and connectivity mode but still rely on both process and artifact relationship information. This enables users to flexibly establish ad-hoc work groups and in addition to have support for the resources they require for collaborative work.

4. Evaluation and Case Study

One of the case study providers is a multi-national company in the market of global telecommunication systems and equipment. This company wants to facilitate the ways in which geographically distributed development units of the company divide their work, communicate and collaborate. The case study especially focuses on the mobile and distributed collaboration support for the process of designing mobile phone software.

The DMC architecture is expected to support new and more efficient ways of working. An important requirement, for example, is to enable employees to find *experts* and their availability in the company. Due to the size of the company, it has many employees that have expert knowledge in a specific domain and this expertise should be exploited more efficiently especially when these people are on the move (or in some other branch of the company). Employees should be able to search and contact experts to solve their problems across many organizational units.

The main DMC architecture evaluation criteria in this case study are: Improvement of production processes by defining and improving the distributed design, development and evaluation processes in mobile phones production; product improvement by effectively supporting the enhancement of Personal Digital Assistant (PDA) products. The DMC prototype, that has been developed, is currently in its user acceptance evaluation phase and detailed results can be reported soon.

5. Conclusions and Future Work

In this paper we described a five-layer software architecture for distributed and mobile collaborative (DMC) systems, which provides *mobility of context* to its group members. This architecture defines a foundation for

the flexible integration of Collaborative Systems (such as Workflow Management, Groupware or Business Process Modeling) with teamwork services that support distributed and mobile collaboration.

Mobility, connectivity and process configuration are based on specified teamwork services that exploit peer-to-peer principles for data sharing supported by client-server structures in contexts of persistency handling.

This DMC architecture enables use cases such as: information sharing and notification of availability (of resources); expert search combined with searching and inviting people for synchronous communication (e.g. chat, video/telephone conference); information retrieval about resources and their profiles (e.g. users, artifacts, processes and their meta-data); community establishment and management.

Future work includes the implementation of additional application-specific services and service configuration facilities based on our current MOTION software system. Results from the currently ongoing end-user evaluation at the industrial partner will be used to further refine the design and integrate it into the prototype implementation.

Acknowledgments

We thank the MOTION team at the Distributed Systems Group: Engin Kirda, Pascal Fenkam, Geri Reif, and Mehdi Jazayeri. This work is partially supported by the European Commission within the project MOTION (Mobile Teamwork Infrastructure for Organizations' Networks) in the IST Program (IST-1999-11400).

References

- [1] K. Aberer, M. Hauswirth, M. Puceva, R. Schmidt, "Gridella: A self-organizing P2P system," Technical Report TUV-1841-01-09, 2001.
- [2] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. Addison-Wesley, Reading, Mass. and London, 1998.
- [3] G. A. Bolcer "Magi: An Architecture for mobile and disconnected Workflow," *IEEE Internet Computing*, May and June 2000, pp. 46 – 54.
- [4] J. Bosch. Design and use of software architectures: adopting and evolving a product-line approach. Addison-Wesley, 2000.
- [5] C. Bussler "Enterprise-wide Workflow Management," *IEEE Concurrency*, 7, 3, 32-43, 1999.
- [6] F. Casati, M. Sayal, M.C. Shan "Developing e-Services for composing e-services," in Proc. CaiSE 2001, Computer Science Lecture Notes, Springer Verlag, pp. 171-186, 2001.
- [7] N. Craven and D.E. Mahling, "Goals and Processes: A Task Basis for Projects and Workflows," in *Proc. COOCS International Conference*, Milpitas, CA, USA, 1995.
- [8] U. Dayal, M. Hsu, R. Ladin, "Business Process Coordination: State of the Art, Trends, and Open Issues" in *Proc. of the 27th VLDB Conference, Roma, Italy, 2001*.
- [9] C.A. Ellis and G. J. Nutt, "Office information systems and computer science," *Comput. Surv.*, vol. 12, no. 1, pp. 27-60, 1980.
- [10] C. A. Ellis, K. Keddara, G. Rozenberg "Dynamic Change within Workflow systems," in *Proc. COOCS International Conference*, Milpitas, CA, USA, 1995.
- [11] C.A. Ellis, S.J.Gibbs, G.L. Rein "Groupware: some issues and experiences," *Communications of the ACM*, 34, 1, 39-58, 1991.
- [12] P. Fenkam, DUMAS: Dynamic User Management System, Master's Thesis, Technical University of Vienna, Distributed Systems Group, June 2000.
- [13] F. Flores, et.al. "Design of Systems for Organizational Communication," *ACM Transactions on Office Information Systems*, 6,2, April 1988.
- [14] Li Gong. "JXTA: A Network Programming Environment," *IEEE Internet Computing*, 5(3):88-95, May/June 2001.
- [15] J.Grudin "Why CSCW Applications Fail," *Proceedings of the ACM CSCW88 Conference*. August 1988.
- [16] A. Hausleitner, S. Dustdar, "Caramba - Ein Java basiertes Multimedia Koordinationssystem", in: *Erfahrungen mit Java. Projekte aus Industrie und Hochschule*. Silvano Maffei, et al. (Eds.), dPunkt-Verlag, Heidelberg 1999.
- [17] M. Jazayeri, A. Ran, F. van der Linden. *Software Architecture for Product Families: Principles and Practice*. Addison-Wesley, 2000.
- [18] E. Kafeza, D. K.W. Chiu, I. Kafeza "View-based contracts in an e-service cross-organizational workflow environment," in *Proc. TES 2001*, Computer Science Lecture Notes, Springer Verlag, pp.74-88, 2001.
- [19] Kirda E., Reif G., Fenkam P., Gall H., Picco G.P., Cugola G., A Web-based peer-to-peer architecture for collaborative nomadic working, 10th IEEE Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), Boston, MA, USA, June 2001.
- [20] P. B. Kruchten. The 4+1 View Model of Architecture. *IEEE Software*, 12(6):42-50. IEEE, November 1995.
- [21] M.P. Papazoglou, M.A. Jeusfeld "Distributed, Interoperable Workflow support for electronic commerce," in *Proc. TREC 1998*, Computer Science Lecture Notes, Springer Verlag, pp.192-204, 1998.
- [22] D.L. Parnas. On the criteria to be used in decomposing systems into modules, *Communications of the ACM*, 15(12), 1053-1058, December 1972.
- [23] D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, 17(4): 40-52, October 1992.
- [24] J. Puustjärvi, H. Laine "Supporting cooperative inter-organizational business transactions," in Proc. *DEXA 2001*, Computer Science Lecture Notes, Springer Verlag, pp. 836-845, 2001.
- [25] T. Schal, *Workflow Management Systems for Process Organizations*. New York: Springer, 1996.
- [26] Workflow Management Coalition (WfMC) "Workflow Management Specification Glossary". 1995
- [27] T. Winograd and F. Flores. *Understanding Computers and Cognition*. Norwood, NJ: Ablex, 1986.
- [28] L.Zeng, B. Benatallah, A.H.H. Ngu "On demand business-to-business integration," in Proc. *CoopIS 2001*, Computer Science Lecture Notes, Springer Verlag, pp. 403-714, 2001.