Architectural concerns in distributed and mobile collaborative systems

Schahram Dustdar and Harald Gall
Distributed Systems Group, Vienna University of Technology
Argentinierstrasse 8/184-1, 1040 Wien, Austria
{dustdar, gall}@infosys.tuwien.ac.at

Abstract

There have been considerable attempts to integrate Workflow Management Systems (WfMS), Groupware Systems, and Business Process Modeling Systems to provide a uniform platform for distributed and mobile collaboration (DMC) of geographically dispersed project teams. Such distributed and mobile teamwork defines new challenges for current IT platforms in terms of architecture and business-specific configurations. This paper discusses architectural concerns for such DMC systems and provides a framework for process aware distributed and mobile teamwork. This is achieved by integrating process- and workspace management requirements with Peer-to-Peer Middleware, Publish-Subscribe, and Community and User Management. The paper discusses a three-layer architecture that integrates process awareness with the easy to use groupware (workspace) metaphor.

Keywords: process awareness, software architecture, mobile collaborative systems

1 Introduction

Software systems such as workflow management, Groupware, process modeling, and project management have been used to automate or to augment business processes in organizations [4, 7, 9, 14, 15, 32]. In recent years there has been considerable attempts to merge or to integrate project management systems, workflow management systems (WfMS), Groupware Systems, and business process modeling systems [12]. Corporate research labs [6, 7] and product teams [2, 22, 26] have made significant steps forward. Future distributed and mobile collaborative systems focus on covering inter-organizational processes (e.g. product value-chains) and their activities on the Internet [4, 10, 25, 28, 30, 31, 34] regardless of location (mobility) and regardless of devices used. A simple sketch of a conceptual architecture for distributed and mobile collaborative systems was presented in [13].

In today's business environments participants in virtual project communities (VPC) demand process awareness to a relatively high degree of the software they use for collaborative work. In addition organizational awareness (e.g. roles) and mobility aspects become increasingly relevant. Current WfMS and Groupware systems do not combine those features virtual project communities need: *information sharing, process sharing, process composition, and process configuration.* Future systems for virtual project communities 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, the team configuration (i.e. participants and their roles or skills), their associated artifacts, and connectivity modes of group members (such as connected, disconnected, or ad-hoc).

Business processes in general and associated workflows in particular exist as logical models. Business Process Management Systems and WfMS complement each other. Workflow systems generally aim at helping organizations' team members to communicate, coordinate and collaborate effectively and efficiently. Therefore WfMS possess temporal aspects such as activity sequencing, deadlines, routing conditions, and schedules. 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 sufficiently support virtual project communities, which require tools for frequent changes regarding process participants, ad-hoc formation of groups collaborating on a business process, and device independent support of group activities. Unfortunately today's WfMS assume that each *work item* is executed by a *single* worker. Hence, distributed collaborative work in virtual

Proceedings of the Eleventh Euromicro Conference on Parallel,Distributed and Network-Based Processing (Euro-PDP'03)
0-7695-1875-3/03 $17.00 © 2003 IEEE

project communities finds only limited or no support by WfMS. Most WfMS focus on automating structured (modeled) intra-organizational business processes. Groupware [16], on the other hand, typically does not contain any knowledge or representation of the *goals* or underlying business *processes* of the group [15, 18, 20, 21, 29].

The contribution of this paper therefore is to elaborate on architectural concerns for process awareness in distributed and mobile collaborative systems. We achieve this by decomposing process - and workspace management issues and presenting a three-layered architecture, which integrates process awareness with the easy to use groupware (workspace) metaphor. The remainder of this paper is organized as follows: Section 2 discusses architectural concerns such as mobility, publish/subscribe, peer-to-peer middleware, and web services. Section 3 decomposes the proposed system architecture and elaborates on the layers and components. Section 4 provides a case study on process composition and process configuration in the setup phase and in the operational phase. Section 5 evaluates the case study and section 6 summarizes the paper.

2 Architectural aspects

An architecture that supports mobility of participants and computers has to be highly flexible and adaptable to new requirements and new collaboration scenarios. In contrast to traditional software architectures, architectures that support mobility are faced with several additional difficulties: because of bandwidth restrictions, unreliable connections and disconnected operations, mechanisms and components are necessary to locate participants, synchronize data and query available resources. Depending on the location of the participant, the number of offered services may vary in quantity and quality. Sophisticated subscription mechanisms and notification services are necessary to disseminate information to the mobile participant instead of forcing the participant to find it. Furthermore, the information in mobile environments needs to conform to different standards such as WML or WAP for data representation. The varying display sizes limit the amount of information that can be displayed on a small mobile device. Thus, components are required that render the information according to the display capabilities of a certain device. High-quality wireless multimedia communications such as UMTS will further improve the quality and quantity of services on the participant's mobile device and have to be considered for a distributed and mobile collaborative (DMC) systems' architecture as well. Much of the required functionality already exists as Web applications, so the key is to efficiently migrate the features to mobile applications. The most common approach is called Web-to-wireless: this relies on existing Web services, but adds a wireless channel to them, providing stakeholders such as customers, employees, and business partners with an access point.

Mobile architectures have to integrate both fixed and mobile components. Hence, we designed the DMC architecture with the following specific design goals in mind. The architecture has to be: *open* with respect to integration of existing technologies and tools; *generic* to be deployed in organizations with varying internal organizational structures, business processes and IT infrastructures; *scalable* for different number of participants, future extensions and new requirements; and *adaptable* to restrictions imposed by mobility both of the mobile participants and the mobile devices.

2.1 Mobility

Process participants are not confined to one location, but may be on the move while working. For this, we consider three connectivity modes in our DMC infrastructure: *connected mode*, *disconnected mode*, and *ad-hoc mode*. Connected mode is used whenever (fixed) network connectivity is available. There is a single global context determined by all hosts on the network. Information can be accessed and shared from any point in the network, at any time.

Disconnected mode is an operational mode, which is required for mobile working when most of the information available in the system cannot be obtained by the occasionally disconnected mobile user. In our architecture, the disconnected mode is a special case of the connected mode and the mobile user is able to continue working (with limited functionality) even if he is disconnected. Once the mobile user is connected again, the changes he has performed locally are synchronized with the rest of the system.

In the ad-hoc mode, mobile users, for example, run a face-to-face meeting and do not have any fixed network connectivity, so they create a temporary network (e.g. in a wireless LAN without Access Point). The context for information sharing and cooperation is limited to the one provided by the parties in communication and to the resources (PDAs,

laptops, etc.) they bring along. Hence, participants can build *ad-hoc* networks for collaboration and information exchange.

2.2 Data Sharing in virtual project communities

We use the notion of a virtual project community (VPC) in the context of work or research, but not as places for social exchange (e.g. persistent meeting rooms). A VPC is, therefore, defined as a persistent technological environment that supports multiple styles of interaction and multi-user engagement. Different groups – including participants on the move – may need to interact in a variety of ways among different levels of a virtual work space.

Sharing data among members of particular VPCs is one essential requirement for a DMC architecture. Participants decide what information they want to share with the members of their community (or even across communities). An architecture for mobile users that supports such a loosely coupled information sharing would typically follow the fully-distributed information sharing principles of Gnutella et al. Such a peer-to-peer communication infrastructure is especially advantageous to support different modes of connectivity. Because every computer (= peer) can work as a server as well as a client [19], it is possible to build ad-hoc networks rather easily.

2.3 Distributed Searches and Data Delivery

To further enhance the effectiveness of virtual project communities, resources are described in meta-data that cover a description of each resource (such as documents, processes, users, communities, etc.) and enable a much more powerful search for information in an enterprise network. People can join communities based on their expertise or interest. Others can also search for information or other people's expertise and be notified whenever this is available in the system (on some peer or server peer). Meta-data is represented in XML and can be searched through, for example, XQL [26]. This allows flexibility in the definition of attributes for meta-data and provides additional extensibility for changing requirements. Artifacts themselves do not need to reside on server peers but can be located on the peer of the community member who shares his document with some community. If meta-data are additionally stored on server peers then such searches even provide results if some peers hosting the particular documents are offline. Notifications and messaging services can be used to ask community members for certain documents or process descriptions in a location independent way since members can be notified by means outside such a process aware framework (e.g. SMS).

Publish/Subscribe mechanisms allow distributing information to where it is wanted. Unlike point-to-point messaging, providers and consumers of information do not need to know about each other. Push and event-based systems are closely related. The purpose of push systems is a timely distribution of data and information to consumers whereas event-based systems focus on notification of events. Both push and event-based systems play an important role in the DMC architecture. We use a push system to notify participants based on a *profile* [23]. The participants define profiles for the kind of information they are interested in, and the push system delivers the information whenever it is available. Participants may subscribe to (and unsubscribe from) specific topics. Furthermore, participants can also subscribe to other participants and be notified whenever these participants are online (or available etc.), which is particularly interesting for mobile collaboration.

2.4 Business Process Support

The scalability and the distributed nature of the Web has made it a popular platform for building collaborative tools. Thus, many Web-based tools have been introduced and there are countless Web applications for improving communication, information exchange and process management. Boeing, for example, has been successfully using Web-based collaborative applications in the construction of its airplanes. To meet the requirements and to cover the scenarios mentioned earlier in this paper, the DMC architecture utilizes the existing Web infrastructure and exploits many of its advantages: Web access is widely available (e.g. on airports), Universal Resource Locators (URLs) are a simple and unique way of identifying resources on the Internet, and Web technologies also offer security against sniffing attacks by using secure HTTP connections through SSL.

3 Architectural components for distributed and mobile collaborative systems

Based on the recent results in software architecture research and practice [1, 2, 3, 11, 24] we adopt the quasi-standard terminology to describe a DMC architecture: A software architecture typically

includes the description of *components*, *connectors*, and *configurations* [1]. 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 three 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 distributed 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 Architectural Units

Before describing each of the components depicted in Figure 1, we group them into logical units with clear responsibilities. A DMC system consists of the following three layers:

- The *Middleware Layer* provides communication means between peers and their software components; it is a communication layer that supports P2P protocols such as Gnutella or P2P architectures such as JXTA [19].
- The *Service Layer* provides the functionalities required for mobile and distributed collaboration: *Basic Services* such as Authentication and Access Control, Resource (i.e. artifact) Management, Process Composition and Configuration, Publish-Subscribe and Distributed Searches as well as *Collaboration Services* such as User and Community Management. The *Collaboration Layer* provides uniform access to all kinds of teamwork services that can be used in applications such as WfMS or Groupware in a DMC context.
- The *Application layer* offers service access and configuration facilities for business-specific services such as running a Design Review or a Production process. It includes process management to configure and instantiate particular business processes in terms of communities, processes, and workflows. Further this layer includes workspace management to assign artifacts and community spaces to project teams.

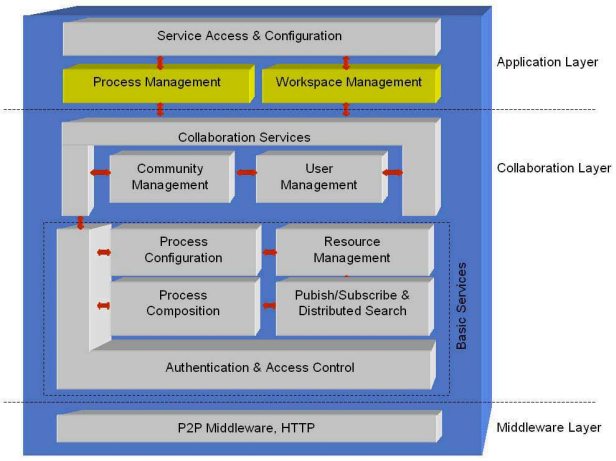

Figure 1 – DMC Conceptual Architecture

3.2 Peer-to-peer middleware

An event-based middleware is used as a *scalable* and *flexible* infrastructure to transfer messages to the participants and components [11] in the connected, disconnected and ad-hoc modes of operation. The middleware exhibits a peer-to-peer architecture (P2P) following the fully-distributed information sharing principles of Gnutella et al. The peer-to-peer communication infrastructure is especially advantageous to support different modes of connectivity. Because every computer (= peer) can work as a server as well as a client, it is possible to build ad-hoc networks rather easily. The P2P middleware is the central underlying component of the system. It manages the subscriptions and provides an event-based system to notify participants on the subscribed events. It is also responsible for managing the virtual communities and for locating the actual physical location of a requested resource.

The middleware is also amendable for queuing participant actions and events that cannot be processed when working in the disconnected or ad-hoc networking mode. Notification messages that cannot be delivered because of an unreachable peer are queued as well. All these queued actions and events are processed as soon as the peer reconnects to the DMC platform (depending on the time-to-live for such events).

3.3 DMC architectural components

In the description of the key components of the DMC architecture we focus on the connectivity and process awareness as basis for WfMS and Groupware systems, which is denoted as *Basic Services* in Figure

1. Users (or process participants) should be granted access via various types of devices ranging from PCs, notebooks to PDAs or mobile phones for connected, disconnected or ad-hoc mode.

Participants can be addressed and reached via the concept of a *community* that resembles a project group. This concept allows building communities for specific purposes and tasks as the basis for 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 and 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 cover 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, 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 Distributed Search is a component that provides loosely coupled communication among components. Its focus is 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. A distributed search, therefore, queries XML repositories (of different content) on each peer and – if successful – returns the requested piece(s) of information. Distributed Searches further allow querying for information that a user wants to be notified whenever it becomes 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) community. This enables the exchange of expertise across communities and processes, which is especially important in mobile and distributed collaboration in 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) [17] and a security component responsible for integrity, confidentiality and authentication. The access control system covers three responsibilities: user control, community control, and authorization.

The above Basic Services components are shielded by the Collaboration Services to provide uniform access for DMC applications. Based on this DMC Service layer any specific DMC application such as WfMS or Groupware can configure the Collaboration Services according to their specific requirements and also build new business-specific services on top of the DMC Services layer. Such a service configuration, therefore, includes the instantiation of processes (templates) and communities (including artifacts, users, and access

rights) for specific tasks (e.g. holding a Design Review while process participants are on the move, in different branches of the enterprise, and/or work on various devices). For more detailed component descriptions we refer to [26].

3.4 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 *connected* 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, HTTP, or RMI. 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 mixture of connected 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 no 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.

3.5 Configurations

Configurations of the DMC architecture depend on the specific business requirements and range from workflow and process to workspace settings. Design Reviews or Production Process Support, for example, can be configured and instantiated company-wise. This includes 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 connectivity modes (from connected to ad-hoc). Virtual project communities can be instrumented in many different ways considering the requirements of the actual organizational unit, the process, and the location: some instrumentations consider the location-aware dimension, i.e. it is of particular interest where the resource actually is residing; others focus on a location-transparency in which it is important that some task is carried out but independently of where the actual resources are. 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.

4 Process composition and configuration

Generally speaking, we distinguish between two phases: the setup phase and the operational phase. Figure 2 depicts a sequence diagram for the setup phase by providing an in-depth analysis of the activities, actors, and artifacts during this phase. The *setup phase* consists of process composition and subsequent configuration.

During *Process Composition*, a *Process Designer* composes a review business process (Design Review Process) consisting of (pre-modeled) *Process activities*. Process type and specifications are selected and results of the selected templates are chosen from a *Process Templates* repository.

In the *Process Configuration* phase, the Process Designer configures a Design Review *Community Manager*. The Community Manager creates the required roles of the *Community Members* and creates the relationships between the previously composed process, the created roles and the artifact templates (e.g. documents, checklists, presentations, etc.). The Community Manager provides those described relationships to the Process Designer. This concludes the Process Configuration activities.

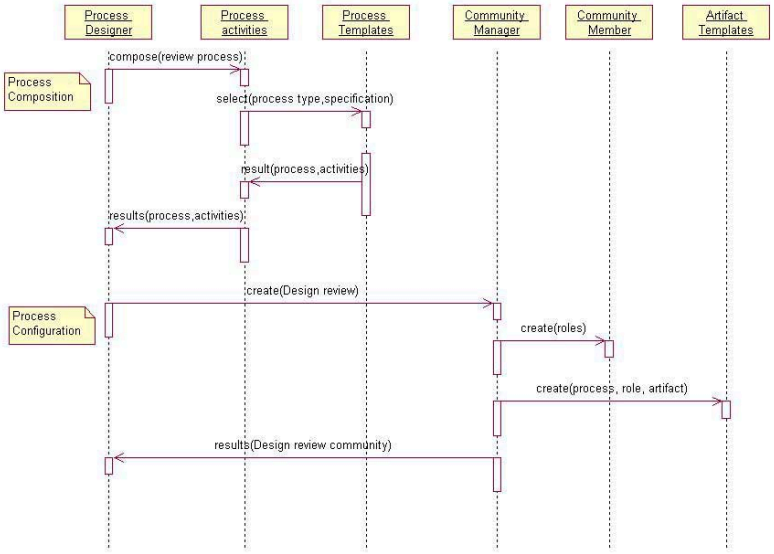

Figure 2. Process Composition and Process Configuration – Setup Phase

Communities act as major conceptual abstraction as depicted in Figure 3: the *Design Review Community* provides an information sharing workspace across peers (for *Project Manager*, *Project Members 1 and 2*, and *External Expert*). The community further works as a context platform for the instantiated process of a design review and supplies the necessary infrastructure for a team (the *Design Review Team*) to jointly execute a work item. Messages and notifications as well as distributed searches (e.g. for an artifact on in "valve design") can be sent via the community to all of its community members. Once a document is published by a community member as, for example, "available" or "updated" then an artifact retrieval is performed using a direct Web connection via the URL that was published (indicated as *download(artifact,URL)* in Figure 3). Such a retrieval can be done rather easily via the Web infrastructure.

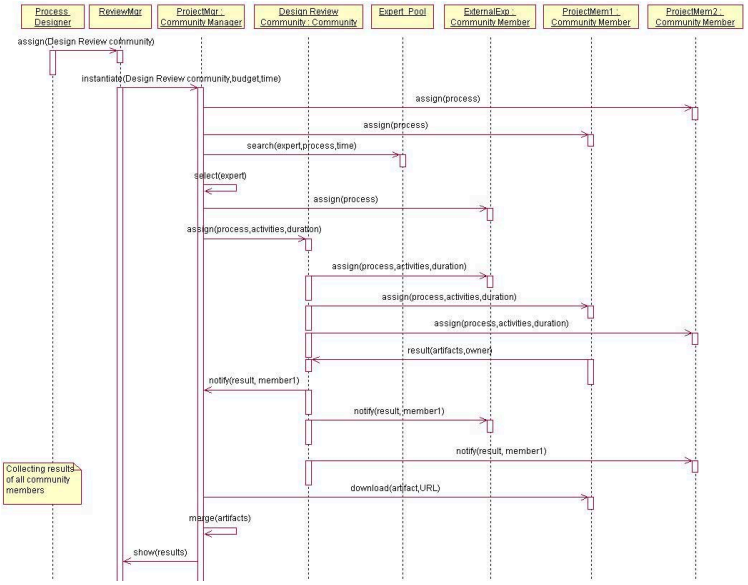

Figure 3. Process Composition and Process Configuration – Operational Phase

5 Usage scenario

One of the case study providers is a multi-national company in the market of household appliance manufacturing. 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 manufacturing processes of certain parts of household appliances.

The DMC architecture is expected to support new and more efficient ways of working. An important requirement, for example, is to enable their employees to find *artifacts* based on a distributed search across enterprise peers. Meta-data representation plays an essential role for this kind of search, distributed searches for artifacts (e.g. using a keyword such as "valve") or for experts (e.g. on "valve design") will open up new possibilities of sharing information of different enterprise units and production domains. Especially when people are on the move (or in some other branch of the company) a business-specific configuration will foster process optimization to speed up collaboration and significantly improve information sharing. The main DMC architecture evaluation criteria in this case study are: Optimization of manufacturing processes by defining and improving the manufacturing processes of household appliances based on effective process composition and configuration exploiting P2P information sharing and distributed searches. The DMC prototype, that has been developed, is currently in its user acceptance evaluation phase with detailed results to be reported soon.

6 Conclusions and Future Work

In this paper we described a three-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), or community establishment and management. Future work includes the implementtation of additional business-specific services and service configuration facilities based on our current prototype software system. Results from the ongoing end-user evaluation at the industrial partner will be used to further refine the design and integrate it into the prototype implementation.

Acknowledgements

The MOTION project is supported by the European Commission in the IST Program (IST-1999-11400). We thank the MOTION team at the Distributed Systems Group: Engin Kirda, Pascal Fenkam, Gerald Reif, and Mehdi Jazayeri.

References

[1] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. Addison-Wesley, Reading, Mass. and London, 1998.

[2] G. A. Bolcer "Magi: An Architecture for mobile and disconnected Workflow," *IEEE Internet Computing*, May and June 2000, pp. 46 – 54.

[3] J. Bosch. Design and use of software architectures: adopting and evolving a product-line approach. Addison-Wesley, 2000.

[4] C. Bussler "The role of B2B protocols in inter-enterprise process execution," in *Proc. TES 2001*, Computer Science Lecture Notes, Springer Verlag, pp.16-29, 2001.

[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] Q. Chen, M. Hsu, U. Dayal "Peer-to-Peer Collaborative Internet Business Servers," HP-Labs Technical Working Paper HPL-2001-14, 2001.

[8] S.J. Chinn, G.R. Madey, "Temporal Representation and Reasoning for Workflow in Engineering Design Change Review," *IEEE Trans. Eng. Manag.*, vol. 47, No. 4, pp. 485-492, Nov. 2000.

[9] 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.

[10] V. Christophides, R. Hull, A. Kumar, "Querying and Splicing of XML Workflows," in *Proc. CoopIS 2001*,

Computer Science Lecture Notes, Springer Verlag, pp. 386-403, 2001.

[11] G. Cugola, E. Di Nitto, A. Fuggetta, "The JEDI event--based infrastructure and its application to the development of the OPSS WFMS," IEEE Transactions on Software Engineering,, 27(9), pp. 827 –850

[12] U. Dayal, M. Hsu, R. Ladin, "Business Process Coordination: State of the Art, Trends, and Open Issues" in *Proc. of the 27th VLDB Confererence, Roma, Italy, 2001*.

[13] Dustdar, S., H. Gall. "Architectural concerns in distributed and mobile collaborative systems", *Proc. of the International Conference on Software and Knowledge Engineering SEKE 2002,* Ischia, Italy, ACM Press, pp. 521 – 522, 2002.

[14] C.A. Ellis and G. J. Nutt, "Office information systems and computer science," *Comput. Surv*., vol. 12, no. 1, pp. 27-60, 1980.

[15] C. A. Ellis, K. Keddara, G. Rozenberg "Dynamic Change within Workflow systems," in *Proc. COOCS International Conference*, Milpitas, CA, USA, 1995.

[16] C.A. Ellis, S.J.Gibbs, G.L. Rein "Groupware: some issues and experiences," *Communications of the ACM*, 34, 1, 39-58, 1991.

[17] P. Fenkam, DUMAS: Dynamic User Management System, Master's Thesis, Technical University of Vienna, Distributed Systems Group, June 2000.

[18] F. Flores, et.al. "Design of Systems for Organizational Communication," *ACM Transactions on Office Information Systems,* 6,2, April 1988.

[19] Li Gong. "JXTA: A Network Programming Environment," IEEE Internet Computing, 5(3): 88-95, May/June 2001.

[20] Groove Networks, 2001, http://www.groove.net

[21] J.Grudin "Why CSCW Applications Fail," *Proceedings of the ACM CSCW88 Conference*. August 1988.

[22] A. Hausleitner, S. Dustdar, "Caramba - Ein Java basiertes Multimedia Koordinationssystem", in: Erfahrungen mit Java. Projekte aus Industrie und Hochschule. Silvano Maffeis, et al. (Eds.), dPunkt-Verlag, Heidelberg 1999.

[23] M. Hauswirth, M. Jazayeri., "A Component and Communication Model for Push Systems," Proceedings of the ESEC/FSE 99 - Joint 7th European Software Engineering Conference (ESEC) and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-7), Toulouse, France, pp. 20-38, Sept. 1999.

[24] M. Jazayeri, A. Ran, F. van der Linden. Software Architecture for Product Families: Principles and Practice. Addison-Wesley, 2000.

[25] 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.

[26] 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.

[27] Klöckner, K. "BSCW: Cooperation Support for Distributed Workgroups", *Proceedings of the 10th European Workshop on Parallel and Distributed Processing PDP2001*, Las Palmas, IEEE Computer Society Publications, Los Alamitos, p. 277-282

[28] R. Krithivisan, A.S. Helal "BizBuilder – An e-Services Framework targeted for Internet Workflow," in *Proc. TES 2001*, Computer Science Lecture Notes, Springer Verlag, pp.89-102, 2001.

[29] P. B. Kruchten. The 4+1 View Model of Architecture. *IEEE Software*, 12(6): 42-50. IEEE, November 1995.

[30] 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.

[31] J. Puustjärvi, H. Laine "Supporting cooperative inter-organizational business transactions," in Proc. *DEXA 2001*, Computer Science Lecture Notes, Springer Verlag, pp. 836-845, 2001.

[32] T. Schal, *Workflow Management Systems for Process Organizations*. New York: Springer, 1996.

[33] Workflow Management Coalition (WfMC) "Workflow Management Specification Glossary". 1995

[34] 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.