Distributed Product Development in Virtual Communities

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

Abstract

Organizations are increasingly forced to manage and coordinate their product and service development processes, to deliver their products and services as fast as possible, and to involve employees, customers, suppliers, and partners seamlessly in different stages of the processes. These processes have to consider that their participants are more and more on the move or distributed while they are working. Expertise needs to be shared across locations and different mobile devices. This paper defines a framework for distributed and mobile collaboration, defines a set of requirements for virtual communities, and discusses a mobile teamwork support software architecture that has been developed in the EU-project MOTION. The framework together with the architecture enables to enhance current collaboration approaches to include the dimension of mobile participants and virtual communities for distributed product development.

Keywords: distributed and mobile collaboration, distributed product development.

1 Introduction

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 Workflow Management Systems (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).

Workflow systems [14] 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 and their process participants [1, 3]. However, traditional WfMS present a rigid work environment consisting of roles and their associated activities and applications.

Virtual project communities require tools for frequent changes of process participants, for ad-hoc formation of groups collaborating on a business process, and for 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 project communities finds almost no support by WfMS. Groupware [3, 6], on the other hand, typically does not contain any knowledge or representation of the goals or underlying business processes of the group [4, 5, 8, 10].

In this paper, we discuss distributed product development in a virtual project community provided by the MOTION teamwork services platform [11, 13]. The contribution of this paper therefore is *a scenario-based discussion of process-aware software engineering in distributed and mobile collaborative systems*. We, therefore, adopt the design review scenario to express process- and workspace management issues and present a layered architecture, which integrates process awareness with the virtual workspace metaphor.

The remainder of this paper is organized as follows: Section 2 reviews the current state of the art and why current systems do not cater for the requirements of distributed product development for virtual teams. Section 3 describes the mobile phone design case study and shows the business requirements for collaboration support. Section 4 discusses the requirements for product development in more detail, analyses the design review phases, their activities and roles as well as the services that were distilled for the teamwork support system. Section 5 shows the software architecture of our software solution. Finally, Section 6 concludes the paper.

2 Systems Support for Virtual Project Communities

Recent advances in the area of Internet Computing and collaborative WfMS are often seen as key technologies for supporting mobile and distributed teams. Cooperative tasks in teams are increasing and as a consequence the use of collaborative systems is becoming more pervasive. To fully understand the context of collaborative technologies for virtual organizations it is important to first analyze the dimensions of current systems.

A business process such as "customer order entry" can be modeled using a traditional WfMS. However a modeled 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 in which constant adaptation to new situations is required and teams are increasingly mobile and distributed. An example of an ad-hoc process is the discussion of a project's requirements document using e-mail (Groupware).

There are two distinct approaches in the knowledge management domain: retrieval or collaboration. The retrieval approach assumes that members of the organization contribute to a "group memory" by submitting/sending documents to a central repository. This repository can be searched (retrieved) and as a consequence be presented as an "Enterprise Portal" accessible with a web browser. The collaborative approach uses the e-mail metaphor and integrates the option to "mail-in" documents into the group memory. Generally one can see the trend of integrating customers into core business processes whenever it is possible (e.g. Customer Relationship Management (CRM) or Partner Relationship Management (PRM)).

Research shows that team performance is positively affected by communications between team members, as shown in [12]. Literature stresses the importance of the formal and informal communicative aspects of collaborative systems, which reflect the underlying structural dependencies in work settings [2, 12]. Working in organizations is often characterized as "networks of commitments," as people in the organization send work through the systems [8]. In this paper, we focus on ad-hoc business processes and on mobile and distributed collaboration for distributed software engineering processes.

In the MOTION project, we have distilled services for distributed product development as described in Section 4.3. These services can be instrumented in many different ways considering the requirements of the actual organization, 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 use the system in a location-transparent way in which it is important that some task is carried out but independently of where the actual resources are.

	Requirements for Virtual Communities			
	Meta-data retrieval	**Expert Search**	**Information Sharing**	**Notifications**
Synchronous Communications (Audio/Videoconferencing)	1	1	3	1
Asynchronous Communications (e-mail)	1	1	3	5
Synchronous Collaboration (Shared Editing Groupware)	1	3	3	3
Asynchronous Collaboration (Workspace Groupware, Intranet)	3	1	3	1
Asynchronous Coordination (Workflow Systems)	3	3	1	3
DMC Systems (Virtual Community Software)	5	5	5	5

Figure 1. Technologies and Features

Legend: full support: 5 limited support: 3 no support: 1

In Figure 1 we summarize our evaluation of technologies supporting teamwork in a classification matrix using a simple scale with three types of support for the requirements we outlined above. Basically we differentiate between synchronous and asynchronous technologies for teamwork support. For each category we provide a well-known example system. During our case study requirements analysis, we came to the conclusion that distributed product development in virtual communities requires a blend of synchronous and asynchronous systems support for communications as well a basic support for asynchronous coordination of team members and their activities.

The requirements for distributed product development teams in virtual communities cannot simply be met by using a combination of traditional synchronous and asynchronous systems since the criteria for successful systems in this area differ substantially compared to traditional "enterprise information systems". We identified and implemented four fundamental feature sets for our case study scenario (meta data retrieval, expert search, information sharing, notifications) and refer to a system having successfully implemented those as a DMC-system (Distributed and Mobile Collaboration).

In our future work we plan to design and implement support for definition, configuration, and composition of processes for virtual communities. In this respect we will integrate process modeling and enactment capabilities in peer-to-peer infrastructures.

The remainder of this paper discusses the design goals of such a mobile distributed collaboration framework, provides a technology mapping from the requirements to a software environment and outlines a particular Web-based peer-to-peer architecture that has been developed in the EU-project MOTION that especially considers mobility of project participants [11, 13].

3 Case Study: Mobile Phone Design

We have investigated the process requirements for the distributed product development of mobile phone software for a large European telecommunications company whose development sites are located in several countries all over the world. Each development site has main responsibilities of some products but the development sites co-operate quite extensively in many phases of the development of the products. The complexity of the product family, the structure of the development organization, and the market pressure in terms of price, performance and rapid need to introduce new features as they become available in the networks make mobile phone software development a challenging task. The company applies two different techniques to cut down product development costs and reduce time to market:

- *Product family architecture:* Common parts of hardware and software of mobile phone generations are developed for the entire family and only product-specific variations or parts are developed in individual product development projects.
- *Concurrent engineering:* Common parts and the product specific parts are developed concurrently in different development sites. Therefore, new products can be introduced to the market at the same time world-wide.

The development of mobile phones software involves many different steps and processes among which are, for example, platform definition, platform management, component development, as well as platform and component integration for product development. In the MOTION project we have performed a detailed analysis of collaboration activities and technologies used for processes such as software development, configuration management, or design review in particular. Project managers of single-site and multi-site projects were interviewed on how they achieve their project goals in terms of collaboration, technologies and processes used. We summarize our findings below, but details of this study are beyond the scope of this paper.

One of the critical aspects in the company's product development is effective, focused, and timely information sharing. In the studied projects it turned out that information sharing is done in a rather traditional way: The most widely used tool for information sharing is e-mail; further, tools such as Lotus Notes, Intranet, and phones were used to communicate within and across project teams. Some of the interviewees remarked that an effective way to gather and share information was to have casual conversations with people, for example, in communication areas or hallways. Personal networking was also ranked high as a medium to acquire filtered and focused pieces of information. Some interviewees noted that personal networking is more effective than Lotus Notes and Intranet to get information because of limited searching capabilities of these tools. In some projects specific information plans were developed to support effective information sharing. These plans were established at the beginning of a project and they described how and what information was to be shared and with whom.

The interviews also showed several problems of information sharing and distribution: Especially product managers indicated that there are clear information sharing boundaries between projects. Often projects do not want to publish all of the requested information, especially if persons who enquire information are not members of a particular group or project. In addition, locating relevant information is difficult and asking via e-mail was considered to be too slow. Product managers see this as a problem because they need overall information of different components from different groups to build up the final product. Another difficulty is locating and finding relevant information effectively.

The business process analysis further investigated the collaboration tools and technologies used so far. It showed that different synchronous and asynchronous means are used: 1) phone and e-mail; 2) Intranet and Lotus Notes; 3) shared network directories; 4) shared work spaces; and 5) videoconferencing.

E-mail is the most frequently used means for information sharing and communication, but it was experienced to have several drawbacks: large mailing lists transform e-mails to chain letters; roles of process participants fade over time; group conversations are quite powerless and usually no decision can be made; the length of e-mails grows fast and lengthy e-mails are hard to read; many messages are received per day, but important messages easily get lost; many messages are saved but usually they get lost in various folders; many attachments are saved as messages instead of storing them into a database such as Lotus Notes; discussions and information sharing is possible only for a small group of participants; effective conversation can hardly be achieved, if at all; and cultural differences are likely to affect the e-mail discussions.

Phone, on the other hand, is rather used when answers are needed quickly. These situations usually occur at the end of projects when time to delivery is short.

Asynchronous groupware such as Lotus Notes is considered to be an important information management tool and most of project data is stored in such systems. Nevertheless, the study pointed out that users of such systems have perceived several shortcomings: finding information or even the right databases is difficult; access rights management for database causes a lot of additional effort; people do often forget to place their documents into the database; the information representation is not formal enough, which makes searches even more difficult for users; the data placement strategies are manifold and usually there is not enough time to update all concerned databases; and some people found it difficult to use mostly due to lack of training.

Intranet is used frequently, because it provides mostly general information and guidelines. But information is hard to find without adequate effective links in the Intranet, old information often is confusing when searching and it Intranet provides almost no support for locating people or expertise in the company.

As a consequence, *shared directories* are still often used. Project members place their documents and other artefacts into commonly shared disk spaces. Since such directories usually are shared within project teams, only little time is spent on searches and information location. But this concept only works effectively for smaller project groups.

Shared editing and whiteboards as provided by tools such as NetMeeting are needed to decrease the distance between distributed project members. These tools are used for reviews, document editing, drawing, concurrent engineering, and application sharing. However, problems with time zones and different devices used remain open.

Videoconferencing, as an additional synchronous communication means, restricts communication to be more formal than in face-to-face meetings. Meeting materials are difficult to supply especially for technical meetings as they are needed for telecommunications product development. Besides technical problems with connections and multicast support, gestures or moods can not be transferred effectively such that face-to-face meetings cannot be fully substituted by this means. Further, ad-hoc meetings are ineffective due to the high effort for setup and connection establishment. The participation is limited and not every group member can participate in the discussion so that valuable comments or remarks are not expressed.

Multi-site project analysis – for project sizes greater 100 participants distributed over more than 7 sites in different countries – further revealed project collaboration habits in working with other people, teamwork, knowledge work and personal work. This multi-site dimension introduced additional challenges related to sharing of project status and news, getting feedback on people's work and finding information.

User requirements are grounded in limited technological capabilities: information, for example, that has been stored in an asynchronous groupware database needs to be accessible for all group members. Since change notifications are weakly developed, change awareness cannot be achieved so that information pull is the most frequently used pattern.

Distributed knowledge access and distribution via personal contacts (i.e. expertise of people in different sites) is an open issue in multi-site collaboration. Roles of project participants and their expertise are invisible and the relevance and context of specific documents is difficult to gather. The need for keeping documents up-to-date across location boundaries raised the importance of shared spaces to check in/out and control work products (versioning, easily accessible data repository).

We have investigated several processes within that telecommunications company: software configuration management, software release management, and conducting peer review meetings for software designs. In the following, we focus on the *design review process* and show its requirements, special needs, and system support through the teamwork services platform that we have developed in the MOTION project. The findings are based on discussions with quality engineers, software release managers, configuration managers and software tool support managers. Among the different instances of design reviews across product development, we distilled common best practices in performing the process in a multi-site dimension.

4 Design Review Phases

A design review process in the telecommunications company is defined in a separate handbook for mobile phone software development, it follows the SEI Software Process Definition Guide and uses the SPICE process model defined by ISO. The particular instance of a peer (design) review in that company is conducted when a work product has been created and checked to be ready for review. The design review team consists of three to six participants (usually from the same development team) each having one or more roles in the design review.

The work product may be distributed to the reviewers in advance for their individual checking prior the actual meeting. During the meeting the author(s) of the document present(s) the work product, walk(s) through it in detail and reviewers give their comments on defects, suggested changes and improvements. These findings are recorded and the work product is improved (by revision or refinement) by the responsible author(s) after the meeting. The reworked artefact is verified again. Measures and statistics are collected and stored for analyzing the review process.

The design meeting has the following goals:

- Evaluate and improve the work product
- Find as many defects as possible
- Consider alternative implementations/solutions
- Educate and exchange knowledge between the review participants
- Collect software engineering data

4.1 Activities, processes, and roles

At the beginning, the particular peer review plan is created by the review leader and then the review meeting is conducted starting with a preparation phase and finishing with a follow-up phase after the review. For that, information concerning the ongoing project is retrieved from project archives including reference information such as checklists. Experts and all participants are selected and the review plan then is distributed to all review participants. The work product under review is also distributed to the review team optionally including known defect items. Required changes are done according to identified defects and time for re-working the artifact is recorded. The review (project) leader is responsible for creating the peer review plan and conducting the review. Software developers and other participating software reviewers act as experts for the process.

In creating the review plan, the project leader and process participants select the work products to be reviewed, identify checklists, define the standards to be used, and establish completion and (re-)review criteria. In conducting the actual peer review those reviewers that are not involved in that specific software development project act as experts for that review process. Further, due to the different locations of the enterprise and people on the move, it is often difficult to find expert reviewers allocating time and readiness for such a review process.

4.2 Requirements for a distributed product development platform

Based on the process description and the rules for carrying out the design review, we distilled particular requirements for a product development platform to effectively support distributed (software) product engineering.

For the preparation of the design review, the project leader and the software author(s) select the reviewers, designate their roles, set timetable, and invite them to a synchronous walkthrough session. These preparation activities take place a week before the actual session. Participants will get notified and are asked to give respond with their availability. The review leader then stores all this information and documentation into the distributed product development platform called MOTION system (at least 3 days before the session). Reviewers will get notified and get access to the documentation (e.g., a URL, access information, and downloadable forms and documents). Reviewers can give their comments on defects, suggested changes and improvements any time before the session and enter them into the MOTION system. A reviewer thereby should be enabled to follow comments of other reviewers. It should be noted that all these activities in the preparation phase should be done asynchronously just using an information space to put together all the required and generated documents.

The holding of the session itself needs synchronous communication among the review participants. At the proposed date the review leader invites (calls) all reviewers to a synchronous session and a session chairman is assigned including someone taking minutes. For that, the MOTION system should support the leader and the rapporteur to manage their work effectively. The software author presents the material that implies a voice connection and the reviewers present their comments (defect items) that may have been earlier attached to the documentation (asynchronously) in the preparation phase or are attached during the synchronous meeting. All defect items and their originating authors have to be managed in the MOTION system. The provisioning and handling of synchronous communication is outside the MOTION platform, but interfaces for a seamless integration are provided.

For the design review follow-up the software author(s) need(s) to rework the work product based on the list of accumulated defect items. The review chairman then checks that all defect items have been integrated and corrected. For the follow-up also the time spent is recorded. The chairman checks and decides about approval of the reworked artifact. In case the result is rejected, a new meeting will be held to clear up the issues.

4.3 Distilled Services for distributed product development

Given the above activities and roles for the design review process and the requirements for a supporting software system such as MOTION, we elicited and defined the following services:

- S1: information updating
- S2: search for expert
- S3: contacting and inviting people
- S3: web research (for enquiries)
- S4: asynchronous information transfer
- S5: synchronous information transfer
- S6: notification of availability
- S7: discussion in a (virtual) community
- S8: virtual (review or expert) community establishment and updating
- S9: archive updating (community information space)

In terms of systems support, 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.

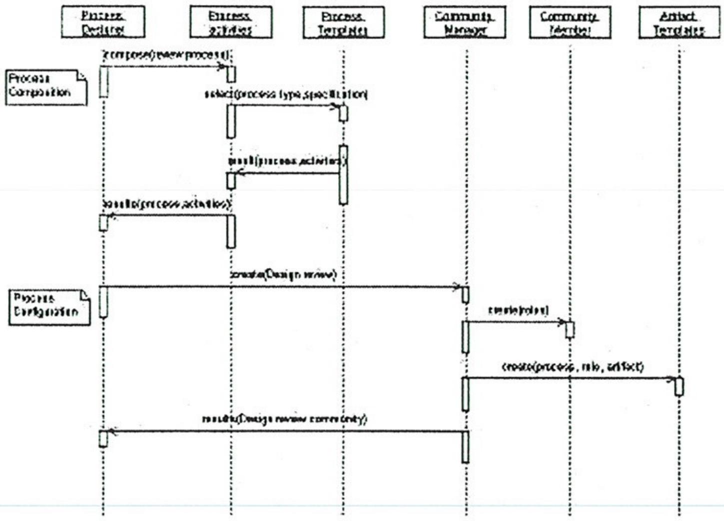

Figure 2. Design Review: Setup Phase

Virtual project *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 "mobile phone 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.

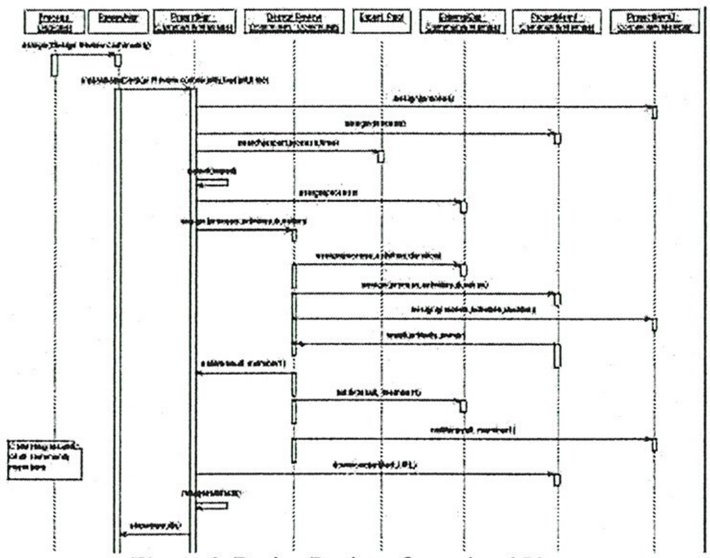

Figure 3. Design Review: Operational Phase

5 Architecture and components for distributed and mobile collaboration

Because an architecture for distributed and mobile collaboration has to cope with three connectivity modes (i.e., connected, ad-hoc and disconnected), we decided to use a peer-to-peer (P2P) infrastructure rather than the classical client/server style. P2P architectures have the main advantage of configurability and scalability. Because there are no dedicated components that only act as servers, a P2P-based system can easily be reconfigured to meet requirements and business-specific changes. Furthermore, the highly distributed nature of a P2P system eliminates the need for business artifacts (that are typically distributed across an organization) to be stored centrally in databases. Storing and indexing business artifacts centrally not only increases the maintenances overhead, but is also not possible in many cases. Although a P2P architecture has many advantages, it also has disadvantages. P2P architectures are more difficult to design than classical client/server architectures because of the increased inter-component communication overhead. The more distributed the components are, the more they need to communicate over a network. This communication overhead can easily become a performance bottleneck. Hence, our DMC architecture has a P2P nature in cases where this is beneficial, but also exploits the classical client/server paradigm where appropriate. The following sections describe the peer-to-peer infrastructure, the main supported collaborative features and the involved components.

5.1 Peer-to-peer middleware

The DMC architecture uses a peer-to-peer middleware that acts as the main communication infrastructure. Messages to users and components are transferred using this middleware. An important feature of the middleware is its ability to deal with the connected, disconnected and ad-hoc modes of connectivity. The P2P style makes the building of ad-hoc networks easier because each *peer* (i.e., node) can function as a client or server based on the configuration choices. Besides providing a P2P infrastructure for starting searches and sending messages, the middleware also provides event-based features for subscribing to and publishing events. Hence, the components of the architecture can also communicate by using events instead of sending messages. The middleware deals with disconnections by queuing event notifications and messages if they cannot be delivered. The messages are delivered whenever connectivity is available and the peer is connected to the DMC platform. A time-to-live value can be set for events in case the peer stays disconnected for a long period of time (e.g., a notebook is not used for two weeks). The implementation of the P2P middleware is a system that supports the P2P protocol such as Gnutella (i.e., its Java implementation - JTella) or P2P systems such as JXTA [9].

5.2 Features and components

The DMC architecture focuses on connectivity and process awareness as a basis for WfMS and Groupware systems. Users require access from a wide range of devices such as desktop PCs, notebooks, PDAs and mobile phones. 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 fashion. In the following sections, we list each collaboration feature and describe the components in the DMC architecture that support it:

5.2.1 Meta-data (profiles)

Each user, community and artifact in the DMC platform has a so-called *profile*. A profile is meta-data about an entity. These profiles describe artifacts, users, processes, or communities in a concise way and represent it in XML. The meta-data in the DMC platform is configurable. The profile information consists of a *system* and a *user* part. The system part is not extensible and stores system-specific information such as the last date of modification and the location of the user. The user part can be defined based on the requirements and needs of an organization. The profiles can be adjusted to contain different elements and descriptors. Setting the profiles 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. are part of profile management. Profile management also includes the management of resources (i.e., business artifacts). Artifacts 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.). The community and user profile information is replicated and distributed across peers. Profiles on artifacts, however, are stored locally on each peer. Each user, hence, has a collection of artifacts on a peer that can be shared with other users. In the DMC platform prototype, we use PDOM, an XML database, for storing, editing and deleting profiles on each peer. A wrapper component on top of PDOM provides the integration of PDOM with the other components of the platform.

5.2.2 Information sharing

Information in the DMC platform is shared by using distributed searches across peers. *Distributed searches* are based on the profiles of users, communities and artifacts. A distributed search, therefore, queries XML repositories (of different content) on each peer and – if successful – returns the requested piece(s) of information. The DMC platform uses the XML Query Language (XQL) to formulate and issue powerful queries across peers. The query is propagated by the P2P middleware and handed to the PDOM component. The results are then sent back. When an artifact of interest has been found with a query, it can be downloaded from the remote peer. This way of sharing information is similar to file sharing in P2P systems such as Gnutella and Kazaa. The query mechanisms, however, are more powerful.

5.2.3 Expert search

Expert search is a key feature provided by the DMC platform. Distributed searches can also 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. Expert searches are a specialized form of distributed searching. The profile information of users are queried with XQL to locate domain experts.

5.2.4 Notification

The *publish/subscribe* component in the DMC provides loosely coupled communication among components. Its focus is on subscription to all kinds of resources such as artifacts, users, communities, processes and access rights. 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, and processes. As a result, this component allows notification of specific activities and can be used for process composition and configuration within or across communities. The publish/-subscribe functionality is realized in the DMC platform using the event-based features of the communication middleware.

5.3 Security

Authentication and access control features are provided by a component called DUMAS (Dynamic User Management System) [7]. DUMAS is a security component responsible for integrity, confidentiality and authentication. The access control system covers three responsibilities: user control, community control, and authorization.

6 Conclusions and Future Work

Solutions for collaborative systems in current working scenarios of large and multi-site enterprises have to consider mobility of users and their devices. The technological advances in mobile computing define new requirements for collaborative systems. In this paper we discussed the issue of mobile and distributed collaboration scenarios in which users have to work together while they are on the move. Finding experts, sharing expert knowledge and corresponding resources led us to the definition of virtual communities that – despite the actual location of a user – foster collaboration across organizational units and processes.

We defined a distributed and mobile collaborative system (DMC) architecture that bases upon a Peer-to-Peer (P2P) middleware, uses publish/subscribe mechanisms for notification and definition of interest of participants, and exploits information push technology to inform about (newly) available resources. Virtual communities function as a basic forum for exchanging artifacts on a P2P-basis, searching for experts in the company (based on people's profile), or notify members of a particular community of tasks, results or availability of resources. Synchronous collaboration on artifacts or synchronous communication can be established in such a virtual community, but is not inherent in the architecture.

The DMC prototype is currently under end-user evaluation in a large, multi-site enterprise that also helped in pinning down the DMC requirements in a business process analysis phase.

Acknowledgements

The MOTION project was supported by the European Commission in the IST Program (IST-1999-11400 Mobile Teamwork Infrastructure for Organizations' Networks). We thank the MOTION team at the Distributed Systems Group for their contributions: Pascal Fenkam, Gerald Reif, and Mehdi Jazayeri.

References

[1] C. Bussler "Enterprise-wide Workflow Management," IEEE Concurrency, 7, 3, 32-43, 1999.

[2] 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.

[3] 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*.

[4] 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.

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

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

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

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

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

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

[11] 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.

[12] Mc Donough, E.F., K.B. Kahn, A. Griffin. 1999. "Managing Communication on Global Product Development Teams," *IEEE Trans. Eng. Manag.*, vol. 46, No. 4, pp. 375-386, Nov.

[13] Reif G., Kirda E., 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.

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