

# Architectural Concerns in Distributed and Mobile Collaborative Systems

Schahram Dustdar

Harald Gall

Distributed Systems Group, Vienna University of Technology

Argentinierstrasse 8/184-1

A-1040 Wien, Austria

++43/1/58801-18414

{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 (P2P) 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, distributed and mobile collaborative systems

## 1. INTRODUCTION

The contribution of this paper 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.

## 2. ARCHITECTURAL CONCERNS

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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SEKE '02, July 15-19, 2002, Ischia, Italy.  
Copyright 2002 ACM 1-58113-556-4/02/0700...\$5.00.

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.

## 3. ARCHITECTURAL COMPONENTS FOR DISTRIBUTED AND MOBILE COLLABORATIVE SYSTEMS

Based on the recent results in software architecture research and practice [1, 2, 3, 4, 6, 9] we adopt the quasi-standard terminology to describe a DMC architecture: A software architecture typically includes the description of *components*, *connectors*, and *configurations* [7, 8]. 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.

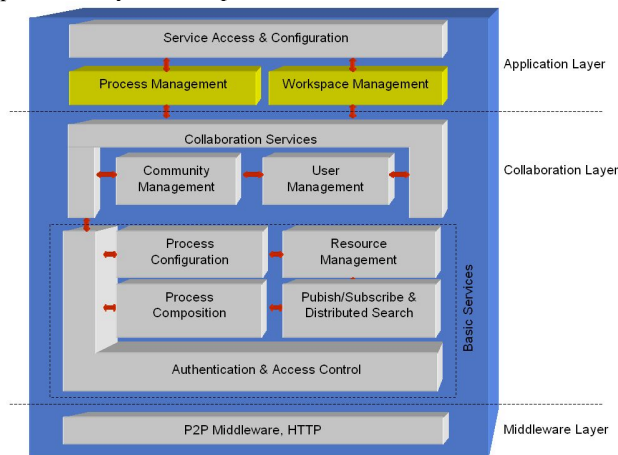


Figure 1: DMC Conceptual Architecture

Before describing each of the components depicted in Figure 1, we group them into logical units with clear responsibilities, following the principles of Parnas [7]. 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 [5]. 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. *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. The P2P 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). 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. *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.). *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). *Process Composition* is concerned with managing process models including coordination and synchronization of its sub-processes and tasks. *Publish/Subscribe and Distributed Search* is a component that provides loosely coupled communication among components. *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.

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. Future work includes the implementation of additional business-specific services and service configuration facilities based on our current prototype software system.

## 4. 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.

## 5. REFERENCES

- [1] Bass, L, Clements, P, and Kazman, R. Software architecture in practice. Addison-Wesley, Reading, Mass. and London, 1998.
- [2] Bolcer, G.A. "Magi: An Architecture for mobile and disconnected Workflow," IEEE Internet Computing, May and June 2000, pp. 46 – 54.
- [3] Bosch, J. Design and use of software architectures: adopting and evolving a product-line approach. Addison-Wesley, 2000.
- [4] Cugola, G., Di Nitto, E., Fuggetta, A. "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
- [5] Gong, L. "JXTA: A Network Programming Environment," IEEE Internet Computing, 5(3):88-95, May/June 2001.
- [6] Jazayeri, M., Ran, A., van der Linden, F. Software Architecture for Product Families: Principles and Practice. Addison-Wesley, 2000.
- [7] Parnas, D.L. On the criteria to be used in decomposing systems into modules, Communications of the ACM, 15(12), 1053-1058, December 1972.
- [8] Perry, D.E. and Wolf, A.L. Foundations for the study of software architecture. ACM SIGSOFT Software Engineering Notes, 17(4): 40-52, October 1992.
- [9] Shaw, M., and Garlan, D. Software architectures, perspectives on an emerging discipline. Prentice-Hall, Englewood Cliffs, NJ, 1996.