A Web-based Peer-to-Peer Architecture for Collaborative Nomadic Working

Gerald Reif*, Engin Kirda*, Harald Gall*, Gian Pietro Picco‡, Gianpaolo Cugola‡, and Pascal Fenkam*

*Technical University of Vienna, Distributed Systems Group
Argentinierstrasse 8/184-1, A-1040 Vienna, Austria
{G.Reif, E.Kirda, H.Gall, P.Fenkam}@infosys.tuwien.ac.at

‡Politecnico di Milano, Dipartimento di Elettronica e Informazione
via Ponzio 34/5, 20133 Milano / Italy
{picco,cugola}@elet.polimi.it

Abstract

With the recent advances in mobile computing, distributed organizations are facing a growing need for advanced Information and Communication Technologies (ICT) that support mobile working. The ability to use information effectively anywhere and anytime has become a key business success factor. Although many Computer Supported Collaborative Work (CSCW) systems have been introduced to date, technologies and architectures that support the collaboration of nomadic workers on a wide range of mobile devices, notebooks and personal computers is still a challenge. The MObile Teamwork Infrastructure for Organizations Networking (MOTION)[1] project is aiming to design a highly flexible, open and scalable ICT architecture for mobile collaboration. In this paper, we present the mobile collaboration requirements of two MOTION industry case studies, and highlight the advantages of a Web-based peer-to-peer architecture and for nomadic working.

Keywords: MOTION, Mobile Teamworking, Information and Communication Technology Architecture, Mobile Computing, Web-based Peer-to-Peer Architectures, Nomadic Working

1 Introduction

Computer Supported Cooperative Work (CSCW) has been a research direction since the early days of computing. Many tools and ideas have been introduced and many commercial products have been implemented (e.g., Lotus Notes, MS Outlook). The main reason why the Hypertext pioneers Engelbart and Nelson started Hypertext research in the 60's was to improve collaboration and working methods by utilizing computer systems [3, 14]. In fact, the Hypertext Markup Language (HTML) and the World Wide Web (WWW) were born because scientists at CERN were looking for ways to improve collaboration over the Internet [1].

Today, many organizations deploy computer-based collaboration tools to improve their productivity and communication. It is not uncommon, for example, that a software engineering project involves over 1000 engineers distributed over 13 countries [2]. Collaborative tools support business processes, information management and knowledge sharing.

With the recent advances in mobile computing, distributed, multi-site organizations are facing a growing need for advanced Information and Communication Technologies (ICT) that provide the ability to access information anywhere and anytime. Supporting collaboration while employees are mobile has been increasing in importance. Employees are not only nomadic users [12], but also *nomadic workers* that would like to continue doing their job while on the move.

Although many CSCW systems have been introduced to date, technologies and architectures that support the collaboration of nomadic workers on a wide range of mobile devices, notebooks and personal computers is still an open challenge. Mobile devices often have small display sizes, limited memory, weak CPU-power, and input limitations due to their size (e.g., writing may be difficult). Furthermore, network bandwidths of mobile devices are often low and disconnections are frequent. Novel technologies, architectures and nomadic working services are required that

[1]This project is supported by the European Commission in the Framework of the IST Programme, Key Action II on New Methods of Work and eCommerce. Project number: IST-1999-11400 MOTION (MObile Teamwork Infrastructure for Organizations Networking)

take these factors into consideration and that can provide effective mobile working and knowledge sharing support.

In the MObile Teamwork Infrastructure for Organizations Networking (MOTION) project, we are developing services based on an ICT architecture to support mobile and distributed collaborative working.

In this paper, we present the mobile teamwork requirements of two industry case studies, and highlight the advantages of a Web-based peer-to-peer architecture and knowledge sharing model for mobile working. The next section lists requirements collected in the case studies. Section 3 presents scenarios based on the requirements. Section 4 presents the concept of knowledge sharing in communities. Section 5 briefly discusses the MOTION architecture. Section 6 presents related work and Section 7 summarizes the paper.

2 Requirements for nomadic team-working

In this section, we give a brief overview of the MOTION industrial case studies and their mobile working requirements.

The first case study is a multi-national company and a large producer of white goods. The company would like to improve the collaboration of its employees in distributed manufacturing activities. The company's manufacturing experts travel around the world and need ubiquitous access for querying distributed knowledge repositories.

Many factories of the company do not have a network or Internet infrastructure. Thus, experts need support to form ad-hoc networks to share documents and exchange know-how on site.

The second industry case study is a multi-national, well-known company in the market of global telecommunication systems and equipment.

The company would like to improve the ways in which geographically distributed development centers of the company divide their work, communicate and collaborate. It would like to increase the availability and working efficiency of its employees by providing support for mobile working, knowledge sharing and collaboration.

Highly specialized domain experts are involved in many research projects and are frequently on the move between the research centers. The developers need support to enable them to search for available domain experts and arrange face-to-face meetings or synchronous communication sessions with them (e.g., using voice- or video-streaming).

The analysis of the case studies led us to the following main requirements for a system supporting nomadic working:

- **Knowledge sharing:** The system has to provide ubiquitous and transparent access to the company's information and services network, from both fixed and mobile nodes, independent from the actual physical location of the user. Updates of the distributed corporate knowledge base have to be processed efficiently. Users may act both as consumers as well as providers of information.
- **Communication:** There is a need for improving interpersonal collaboration and communication among the different parties within the process network. Beside instant messaging this includes support for advanced synchronous communication such as voice- and video-streaming.
- **Device-independence:** The system has to support a wide range of mobile and fixed computing hardware such as Personal Digital Assistants (PDAs), Wireless Access Protocol (WAP) [16] enabled cellular phones, palmtops, notebooks and desktop personal computers.
- **Mobility:** The system needs to support three modes of connectivity for nomadic working: *connected mode*, *disconnected mode* and *ad-hoc mode*.

  In the connected mode, connectivity is strong and a direct connection to the Internet is available from a fixed network node. The user might be at her office, for example.

  In the disconnected mode, the user does not have any network connectivity. She may be in an airport, for example, waiting for a flight. Although the user is not connected, she is able to continue working. The number of tasks she can do offline depends on the information that is available to her on the particular device. Once she is re-connected, she is able to synchronize the changes with the rest of the system.

  In the ad-hoc mode, the user has connectivity that is weak and not optimal. Nevertheless, the user is able to communicate and build ad-hoc networks with devices that are in communication range to form a working community.
- **Subscription and notification:** In a typical collaborative working process, employees often depend on work done by others. They frequently have to wait for some kind of event to happen before they can continue their work. For example, when writing a joint documentation for a software component, a software engineer might need to know when a colleague has released or updated a certain part of the document.

  Hence, subscription mechanisms are needed that enable a user to subscribe to business events and to receive notifications. Notification on the availability of other users is also required for communication and information exchange purposes. The user should be able to receive these notifications also via other means such as SMS or email.

3 Nomadic working scenarios based on the case studies

In this section, we sketch four nomadic working scenarios based on the requirements we listed in the previous section. The scenarios illustrate the interaction of the nomadic worker with the MOTION system and anytime, anywhere access to information.

Scenario 1: MOTION access via WAP

A software engineer, Dr. X, is traveling from Austria to a research center in the US. Her plane is late and she has to wait at the airport for two hours. She takes her WAP-enabled cellular phone, dials up and connects to the MOTION platform. She first goes through her MOTION messages about business events she has subscribed to. She sees that a colleague has updated a research paper they had been working on. She reads a message from a project worker who is looking for an expert in software testing and acknowledges her availability for a chat on the following day. Dr. X updates her personal availability information that she is reachable via SMS for the next two hours and disconnects.

Scenario 2: MOTION access and retrieval via Web

Dr. X has a connecting flight at New York JFK airport. While she is reading a report, she is stuck with a problem. Unfortunately her WAP enabled cellular phone does not work in the US. But she can use a Web terminal at the airport. She opens the web browser and connects to the system. Due to the time zone difference, she does not find any colleagues online so she formulates a query with the required information and disconnects. While disconnected, the search is performed and the result is presented to Dr. X the next time she connects to the system.

Scenario 3: Connected and disconnected working

During the flight to her final destination, Dr. X uses her notebook to work in disconnected mode. She updates the report she is going to present and answers some messages.

When she arrives at the research center in the US, she finds an Internet connection. She boots up her notebook and connects to the MOTION system. After she has logged in, the messages she had written in disconnected mode are sent and her colleagues receive a notification that she has updated the report.

Scenario 4: Ad-hoc networking

Dr. X meets her colleagues at the construction site of a new plant. They wish to discuss their experiences about how to run the new plant efficiently. Until now no Internet connection and network infrastructure is available. Dr. X and her colleagues take their notebooks and form an ad-hoc network using their wireless Ethernet cards. They form a working community and are able to share their artifacts.

4 Knowledge sharing

Knowledge sharing is a key requirement in MOTION. The system has to provide simple access and manipulation mechanisms for the distributed knowledge base. The knowledge base is distributed over several *peers* in the MOTION architecture. In MOTION a peer is any computing device that runs the MOTION middleware. In addition, peers can be in disconnected or ad-hoc mode and therefore not reachable. However, users should be shielded from the complexity of dealing with the actual location of a requested artifact (location transparency).

This goal is achieved by introducing the notion of *community*. A community in MOTION is a set of users, the *community members*. The members are grouped through some membership relation, e.g., a common interest in the design of the latest cellular phone. Each user may belong to one or more community.

Each user in the system owns artifacts that are stored in the user's *resource space*. If a user wishes to share a set of artifacts with her colleagues, she makes them available to a community. The subset of artifacts from the resource space that is made available to a given community is called the *member space*. The member space contains all the artifacts a user wishes to share with a given community. An artifact can also be made available to more than one community. For instance, this paper could be of interest for the community "Software Engineering" and the community "Web Systems". That is, a user can have more than one member space that may overlap.

The set of artifacts the connected community members contribute to a community is called the *community space*. The community space is therefore the union of the member spaces of all community members that are currently connected. From the user view, the access to the artifacts in the community space is transparent regardless of the actual physical location of the artifact. The resource and community spaces have essentially similar basic functionalities that enable the user to query and manipulate the content of the space and to subscribe to occurring events.

The idea of our community space is a dynamic one. Since the community space is built only by the member spaces of the community members that are *currently* connected, the content of this space will change dynamically according to the users connected to the system. In particular, a given resource will be available to the connected community only as long as its owner is part of it. Nevertheless, the ability to communicate and share artifacts asynchronously is a key requirement for MOTION. This requires that specific artifacts have to be accessible persistently to the users connected to the system. For this, a portion of the community space is actually persistent: we refer to this part of the community space as the *community cabinet*. Since the community cabinet is part of the community space, the cabinet is always available to all community members. Thus, an artifact stored in the community cabinet is available to the community members regardless whether the owner of the resource is currently connected.

The concept of MOTION communities and cabinets is a flexible concept for knowledge sharing in a distributed environment. Access control is an important issue in every knowledge sharing system. MOTION uses a flexible user access control and user management component [4].

5 MOTION Web-based peer-to-peer architecture

In this section, we give a brief overview of the MOTION Web-based peer-to-peer architecture that supports the knowledge sharing model and the scenarios presented in the previous sections.

5.1 The Web as the MOTION service platform

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 [5].

To meet the requirements and to cover the scenarios collected in the case studies, the MOTION architecture utilizes the existing Web infrastructure and exploits many of its advantages.

The Web is well-known to the user and a large majority of users are familiar with Web related issues and terms such as Universal Resource Locators (URLs), bookmarks, plug-ins etc. Web clients are available for a wide range of operating system. Furthermore, Web access is widely available (e.g. at airports), but other Internet infrastructures such as telnet or news are not as often supported by public Internet access points. Even if they are supported they are not secure.

The MOTION system has to support synchronous communication such as voice- and video-streaming. These technologies are already supported by a wide range of existing plug-ins and applications. These programs can be integrated into a Web environment.

Universal Resource Locators (URLs) are a unique way of identifying resources on the Internet. Web browsers have integrated functionality for supporting Internet protocols by using URLs. Thus, URLs are well-suited for referencing information sources. Web technologies also offer security against sniffing attacks by using secure HTTP connections trough SSL. The communication is encrypted and is thus secure against eaves dropping.

Because multipurpose publishing [9] and device-independence [8] have increased in importance, standards such as the World Wide Web Consortium's [15] eXtensible Markup Language (XML) and eXtensible Stylesheet Language (XSL) have been defined. As shown in [7], it is possible to utilize these standards and build flexible Web user interfaces that support stationary as well as mobile computer devices.

5.2 Peer-to-peer middleware

The MOTION system requirements for mobility implicate frequent disconnections and changes of the network topology. To meet these requirements the MOTION architecture is based on a peer-to-peer middleware.

The MOTION middleware is a central component of the communication infrastructure. The middleware manages the subscription and provides event-based mechanisms to notify users on the subscribed events. It is also responsible for locating the actual physical location of a requested artifact. The MOTION middleware is implemented as a servlet running on the Web server on every MOTION peer.

The middleware is also responsible for queuing the user 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. All queued actions and events are processed as soon as the peer connects again to the MOTION platform. This process is transparent to the user as described earlier in Scenario 3. The middleware exploits the experiences gained in the LIME [11] project.

We mentioned before that the community cabinet in MOTION guarantees that the artifacts stored in it are persistent and accessible at all times. However, how does one provide persistency in a peer-to-peer environment? In MOTION, a specific set of peers are always up and running. These peers are the *backbone* of the MOTION system and are called *backbone peers*. The backbone peers offer services which are accessible persistently by all the other peers. The advantage of this approach is that any peer can be designated

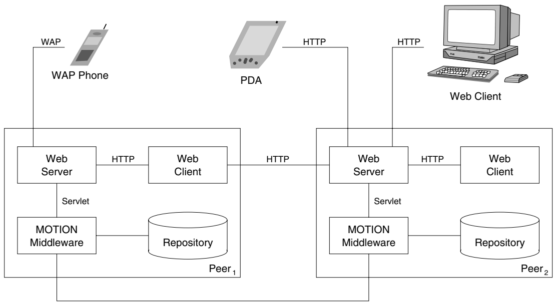

Figure 1. MOTION Architectural Sketch

as being a backbone peer and vice versa. The configuration of the system is highly flexible.

5.3 MOTION Web Architecture

In this section, we provide an overview of the Web-based MOTION peer-to-peer architecture. Figure 1 illustrates typical MOTION peers and their components. It also shows the access to the MOTION system from devices not running a Web-server and the MOTION middleware (Web-terminal, WAP phone, PDA).

Each MOTION peer contains both, a Web server, running a Java servlet engine and a Web client. The middleware provides an API to connect to the knowledge repository. This API consists of generic functions to manage the artifacts stored in the repository. This architecture enables the support of various kinds of repositories such as XML-databases, SQL-databases, file-systems, etc. Only an adapter between the repository and the MOTION API has to be provided. The repository not only stores the actual artifact, but also XML metadata. The metadata is used for managing and querying the repository.

Not every MOTION peer needs to contain the whole machinery depicted in Figure 1. Depending on the computational power and the memory capacity, some components might offer reduced functionality. These constraints reduce the functionalities offered to the user. Due to memory limitations, a MOTION peer on a PDA, for example, cannot host a full SQL-database as a repository. Instead the repository on a PDA could just cache the XML metadata of the retrieved artifacts and access the actual artifact using the URL.

MOTION also supports access from devices not running a web server and the MOTION middleware. These devices access the platform via the web server of a MOTION peer. The only requirement for these devices is that they run a Web- or WAP-client. This enables the user to access MOTION from any computer running a Web browser (e.g. in an Internet cafe), from a PDA or WAP-enabled cellular phone.

Because of the limited capabilities of mobile devices the user interface and the displayed content have to be adapted accordingly. In MOTION we adapt the content using an extended version of the MyXML technology [7].

5.4 Repository access scenarios

In this section, we show the artifact access to the local repository as well as to the repository on a remote peer.

Artifacts that are stored in the local repository are retrieved through the local web server, which delegates the request to the underlying MOTION middleware, which in turn queries the local repository. The retrieved artifact is then transfered to the Web client via the middleware and the Web server. This scenario is illustrated by the sequence diagram in Figure 2.

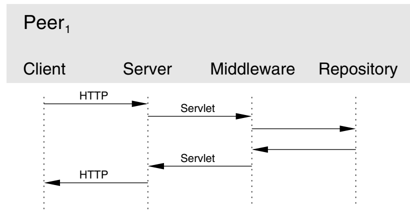

Figure 2. Local access to artifacts

Artifacts that are not stored in the local repository are accessed through the local Web server and the MOTION middleware. The middleware is then responsible for locating the artifact among all connected peers. The URL of the requested artifact is provided to the Web server that forwards it to the browser. The actual artifact is retrieved via a conventional client/server access using HTTP. In this case, the requesting peer operates as client and the peer hosting the requested artifact as server. Since every peer consists of a Web server as well as a Web client every peer can operate in both roles. This scenario is shown by the sequence diagram in Figure 3.

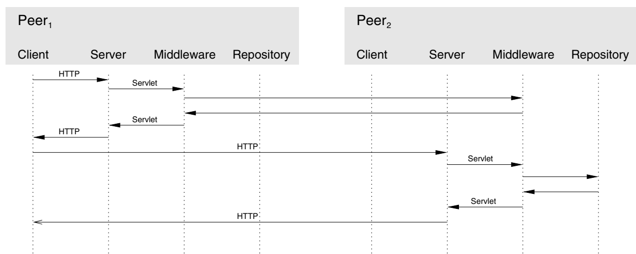

Figure 3. Remote access to artifacts

Proceedings of the 10th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '01)
1080-1383/01 $10.00 © 2001 IEEE

6 Related Work

To our knowledge, not much service and architectural work exists in supporting and improving mobile collaboration, knowledge sharing and working.

The majority of research on architectures supporting mobile computing is concerned with low-level problems found in mobile environments such as resource discovery and low bandwidth. Liu, for example, introduces a virtual system architecture in [10] for supporting wireless and mobile communications that uses predictive mobility management algorithms to predict users' movements within the system. Resources are brought to the user before the user arrives and hence, efficient access to information is provided. The architecture, though, is not concerned with collaboration and information sharing issues and does not provide support for ad-hoc connectivity.

The most notable project with similar goals to MOTION is StudySpace [13]. Although StudySpace tackles problems such as determining network, hardware and display capabilities before fetching a document, it does not address ad-hoc information sharing and community support issues.

A tool that has recently been developed and that addresses some teamwork requirements we have discussed in this paper is Groove [6]. Groove uses a peer-to-peer approach, supports knowledge sharing and enables synchronous communication. But it does not support mobile working and lacks a sophisticated access control system.

7 Conclusion

Supporting collaboration and nomadic working on a wide range of computing devices is a challenge for application developers. Traditional distributed computing concepts, fixed-node networks, services and architectures are not sufficient. Our ongoing work in the MOTION project tackles these problems and designs and implements services based on a generic architecture for mobile working, collaboration and knowledge sharing.

From the case studies we observed a need for ad-hoc networking support and easy sharing of knowledge. Our architecture is based on a dynamic community model and utilizes and extends the existing Web infrastructure to enable organizations to combine the advantages of the Web with the advantages of mobile computing.

We are currently working on the implementation of prototypes of the services and the peer-to-peer middleware. The architecture will be deployed and evaluated in the case study organizations.

A Web-based peer-to-peer service architecture is a promising novel approach to answering the question of how collaboration and knowledge sharing can be improved within and between organizations.

References

[1] T. Berners-Lee, R. Cailliau, A. Loutonen, H. F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37(8), August 1994.

[2] L. Borrmann and F. N. Paulisch. *Software Architecture at Siemens: The challenges, our approaches, and some open issues*, pages 529–543. in Software Architecture, Editor: Patrick Donohoe, Kluwer Scientific Publishers, 1999.

[3] D. C. Engelbart. Toward augmenting the human intellect and boosting our collective iq. *Communications of the ACM*, 38(8):30–3, August 1995.

[4] P. Fenkam. Dynamic user management system for web sites. Master's thesis, TU Vienna, Institute for Software Technology, 2000.

[5] S. L. Fowler, A.-M. J. Novack, and M. J. Stillings. The evolution of a manufacturing web site. in Proceedings of The 9th International World Wide Web Conference, Amsterdam, Netherlands, May 2000.

[6] GrooveNetworks.
http://www.groove.net.

[7] C. Kerer and E. Kirda. Layout, content and logic separation in web engineering. in Proceedings of the 9th International World Wide Web Conference, 3rd Web Engineering Workshop, Amsterdam, Netherlands, May 2000.

[8] E. Kirda, M. Jazayeri, C. Kerer, and M. Schranz. Experiences in engineering flexible web services. *IEEE Multimedia*, January - March 2001.

[9] H. W. Lie and J. Saarela. Multipurpose web publishing: Using html, xml, and css. *Communications of the ACM*, 42(10), 1999.

[10] G. Lui, A. Marlevi, and G. Q. M. (Jr.). A mobile virtual-distributed sytem architecture for supporting wireless mobile computing and communications. *Communications of the ACM*, (2):77–86, 1996.

[11] G. P. Picco, A. L. Murphy, and G.-C. Roman. LIME: Linda Meets Mobility. In D. Garlan, editor, *Proceedings of the 21st International Conference on Software Engineering (ICSE'99)*, pages 368–377, Los Angeles, CA, USA, May 1999. ACM Press.

[12] J. Schiller. *Mobile Communications*. Addison-Wesley, Reading, Mass. and London, 2000.

[13] J. L. Schnase, E. L. Cunnius, and S. B. Dowton. The studyspace project: Collaborative hypermedia in nomadic computing environments. *Communications of the ACM*, 38(8):72–3, 1995.

[14] J. B. Smith and S. F. Weiss. Hypertext. *Communications of the ACM*, 31(7):816–19, July 1988.

[15] W3C.
http://www.w3.org. Technical report.

[16] W3C.
wap forum - w3c cooperation white paper
http://www.w3.org/tr/note-wap. Technical report.