Web services for Groupware in Distributed and Mobile Collaboration

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

Abstract

While some years ago the focus of many Groupware systems has been the support of "Web computing", i.e. to support access with Web browsers, the focus today is shifting towards a programmatic access to "software services", regardless of their location and the application used to manipulate those services. Whereas the goal of "Web Computing" has been to support group work on the Web (browser), Web services support for Groupware has the goal to provide interoperability between many groupware systems. The contribution of this paper is threefold: (i) to present a framework consisting of three levels of Web services for Groupware support, (ii) to present a novel Web services management and configuration architecture with the aim of integrating various Groupware systems in one overall configurable architecture, and (iii) to provide a use case scenario and preliminary proof-of-concept implementation example. Our overall goal for this paper is to provide a sound and flexible architecture for gluing together various Groupware systems using Web services technologies.

Keywords: Web services, Groupware, Distributed and Mobile Collaboration

1 Introduction

Since the late 1960s Groupware aims at supporting various group activities of individuals embedded in multiple teams within organizations as well as between organizations. While some years ago the focus of many Groupware systems has been the support of "Web computing", i.e. to support access with Web browsers, the focus today is shifting towards a programmatic access to "software services", regardless of their location and the application used to manipulate those services. Web services (should) provide the required standards, protocols, and technologies to fulfill this goal. Whereas the goal of "Web Computing" has been to support group work on the Web (browser), Web services support for Groupware has the goal to provide *interoperability* between many groupware systems.

Web services can be seen as a newly emerging distributed computing model for the Web. The standardization process is driven by the growing need to enable business-to-business (B2B) interactions on the Web. Web services are self-contained, self-describing modular applications. The Web services model develops a componentized view of web applications and is becoming the emerging platform for distributed computing. The architecture considers a loosely integrated component model, where a Web-Service interface (component) encapsulating any type of business logic is described in standardized interface definition language, the Web services Description Language (WSDL) [8]. Web-Service components interact over XML messaging protocol and interoperate with other components using the Simple Object Access Protocol (SOAP) [7]. Many software vendors and a plethora of "standardization consortia", e.g. ebXML [3] W3C [7, 8], OASIS [6], are providing models, languages, and interfaces for the *life cycle* of Web services: describing, publishing, unpublishing, discovering, and making them available to users for invocation.

Groupware systems have the potential to offer and consume such services on many levels of abstraction. Consider a typical scenario of team work: (Distributed) Team members collaborate by using messaging systems for communications. In most cases the "workspace" metaphor is used for collaboration. This means that team members have access to a joint workspace (in most cases a shared file system), where files (artifacts) and folders may be uploaded and retrieved. In many cases (mobile)

experts are part of such teams and their workspaces. One can argue that a workspace can be seen as a community of team members working on a shared project or towards a common goal. The aim of Groupware systems is to provide tool support for communication, collaboration, and to a limited extent, for coordination of joint activities. Figure 1 illustrates this scenario.
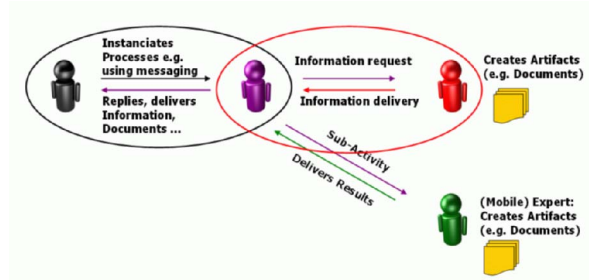

Figure 1. Groupware Communities and Actors

The *contribution* of this paper is threefold: (i) to present a framework for analyzing three levels of Web services for Groupware support, (ii) to present a novel Web services management and configuration architecture with the aim of integrating various Groupware systems in one overall configurable architecture, and (iii) to provide a use case scenario and preliminary proof-of-concept implementation example. Our overall goal for this paper is to provide a sound and flexible architecture for gluing together various Groupware systems using Web services technologies.

The remainder of this paper is organized as follows: Section 2 presents a novel approach for Web services management architecture based on three levels: business, application, and Teamwork service. Furthermore it provides an overview on our suggested Web services management and service configuration architecture. Section 3 discusses the Groupware support provided by Groove workspaces and shows how the provided Web services interfaces can be utilized for the purpose we present in this paper. Finally, section 4 concludes the paper.

2 Web services Management Architecture

Web Services can be used to address several management aspects:

+ business management, i.e. exposing application functionalities as Web services to other business partners
+ application management from a business perspective, i.e. wrapping functionalities provided by business applications under a common service interface [10]
+ service management, i.e. providing common service interfaces to applications for monitoring their operation.

In our case it is essential that Groupware systems such as Groove [5], Caramba [2], Colligo [9], etc. have a common way of interoperability. Tools are still closed with respect to integrating their functionality for a particular business. So, for example, it is not possible to have one team member work with Groove the other one working with Caramba or some other Groupware tool and commonly share information and collaborate on a task across specific Groupware systems. The utmost common denominator is to exchange data via some common file formats (e.g. XML or XGL) but without any business process support. So virtual teams are restricted to particular tools and tool-specific workspaces or formats.
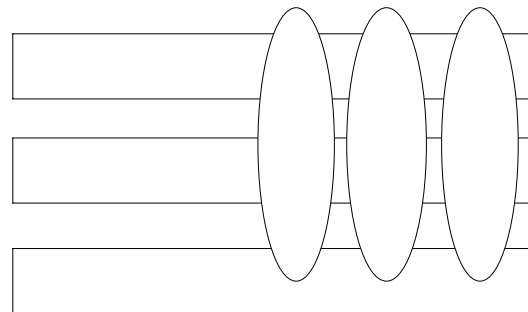

Figure 2. Web services Levels

Web services for Groupware systems for the first time allow such a multi-tool collaboration in the sense of the above mentioned application management. Uniform Web service interfaces would allow to access Groupware specific services such as groups, member data, files, calendars, etc. and share these data for a "higher" business value.

For that, we propose the following Web services management and configuration architecture including particular Web service configuration points (see Figure 3):
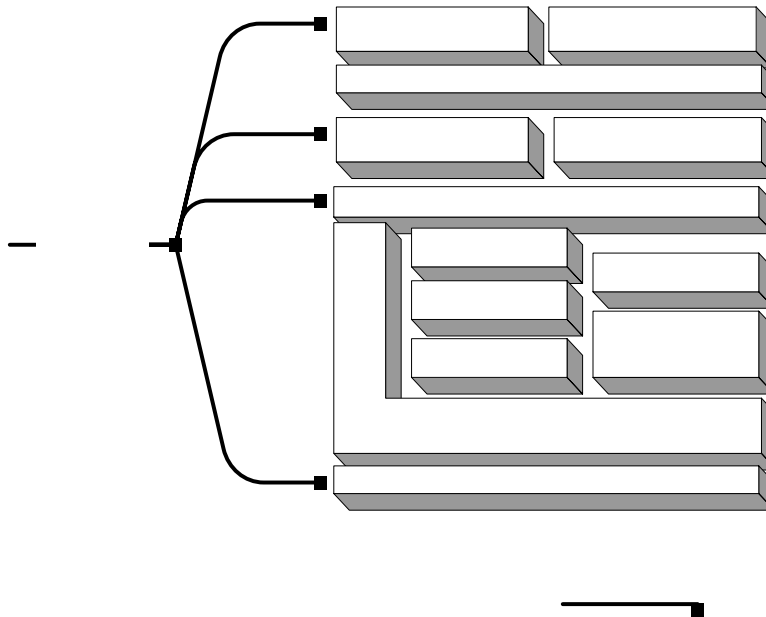
Figure 3. Web services Management and Configuration Architecture

Service
Configuration

In the architecture depicted in Figure 3 we base upon our previous work in which we devised an architecture for distributed and mobile collaboration presented in [1]. There we proposed a teamwork services layer that offers all kinds of *basic services* such as user and group, artefacts, access rights management, distributed searches, publish-subscribe, messaging, or process management.

The API of these Teamwork Services has been enhanced to a Web service interface to allow usage of such services for all kinds of collaborative systems. For example, basic services such as group management realized separately in every Groupware system could then be used in a uniform way. This would exploit the Teamwork Services layer and provide a more general layer as known from communication middleware (e.g. Corba). For Groupware systems, the Teamwork Services will act as a *teamwork middleware* providing the required abstractions and mechanisms for distributed and mobile collaboration scenarios.

As depicted in Figure 3, we propose several Web service configuration points:

+ configuration of applications for business goals (in the sense of business management);

+ configuration of application-specific services, for example of Groove (in the sense of application management); and

+ configuration of teamwork-specific services of a teamwork middleware (in the sense of service management).

These Web service configuration points allow multi-level customization of Groupware applications. The common denominator of such an approach is the Web service interfaces that enable collaboration across Groupware applications but do not restrict application-specific and, therefore, tool-specific feature sets. As long as each tool maps its features to the teamwork services level, the application (interoperability) level and the business level (for activity-oriented collaboration), the room for unique selling propositions is not limited, but, on the contrary, significantly expanded. Interoperability is no more limited to exchanging data in a common file format, but extended to full-fledged inter-tool collaboration.

In the following section, we describe the Web service support that has been integrated in Groove and, as a first significant step, allows to address service integration on the application management level.

3 Groupware support using Groove

Groove Workspace [5] is a desktop Groupware software supporting virtual workspaces for working with different groups of people. Groove allows to manage projects, share files and jointly work on files, discuss work in real-time (e.g. audio-conferencing), and to share presentations. Figure 4 depicts a screenshot of a "PDP2004" workspace, presenting shared files and folders. Each workspace may contain many "tools" to manipulate artifacts of the workspace. Such tools include "Files", Calendar" and a "Discussion" space, where ideas can be structured and (re)grouped (e.g. in brainstorming sessions). More tools can be integrated by a plug-in concept. However, all plug-ins remain proprietary in the sense that only Groove itself has access to the features.
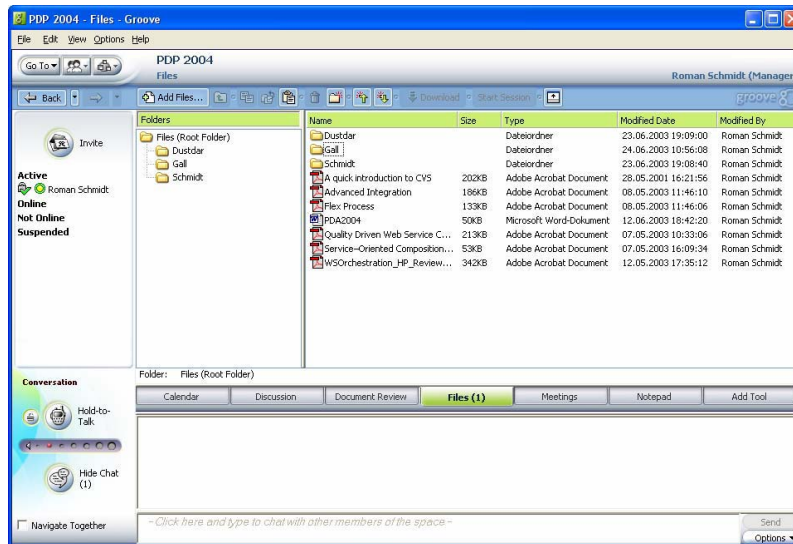
Figure 4. Groove Workspace

3.1 Groove Web services

Furthermore, Groove Web services extend the reach of current Groove tools by providing a way to distribute, access, and process Groove workspace data for customize Groupware systems, by utilizing standard Web services protocols. Groove Web services allow exposing Groove objects and data as a Web service, which makes it easier to provide solutions that work in an extended environment. For example, Groove Web services can be used to:

- Integrate Groove tools with external applications running locally on the same device as Groove or running on a server on the network.
- Provide an integrated solution powered by a Groove tool that runs on any end-point in an IP network, including end-points running any operating system and lightweight end-points that include a SOAP client. In addition, Web access to solutions can be provided.
- Allow a Groove user to access their data when they do not have access to a Groove client by using a Web browser or lightweight SOAP client, such as a cell phone.
- Provide access to data stored in Groove on a Web page on an internal or external Web site.

3.1.1 Groove Web services Architecture

There are three major components that are part of Groove Web services:

- A SOAP client that consumes Groove Web services
- A Groove Web services Access Point
- A Groove client with Web services enabled.

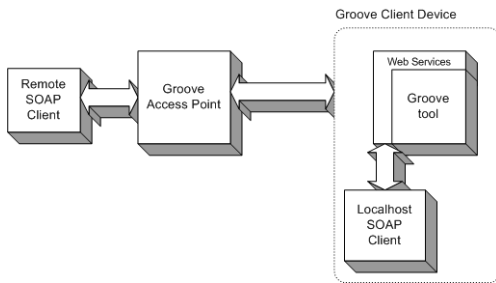Figure 5 illustrates these components and the relationship between the SOAP client and the Groove tool.

Figure 5. Groove Web services Architecture

The remote SOAP client uses the Access Point to connect with Groove clients, the Access Point name registration to identify Groove clients and Groove identities, and polls the Access Point for events fired by Web services. The local SOAP client connects with Groove via "localhost", accesses the accounts available on the local Groove client, and polls the Groove client via "localhost" for events fired by Web services. The Groove Access Point transmits data between SOAP client and Web services exposed by a Groove client, provides name registration for SOAP client and a Groove client, and provides queuing for the SOAP client. The Groove client contains Groove data and objects, exposes Web services, provides access to data and generates events when data changes, and provides a local access point for SOAP clients running on the same device as the Groove client.

3.1.2   Groove Web service Development Kit

The Groove Web services Development Kit (GWS GDK) allows developing SOAP clients accessing Groove Web services locally or remotely using the Groove Access Point. The GDK includes WSDL definitions of all available Web services, tools and sample implementations of SOAP clients and documentation. The included tool Groove Explorer demonstrates the usage of some Web services. It retrieves information about Identities, Contacts, Shared Spaces, and Tools of a local or remote Groove Workspace, as depicted in Figure 6.
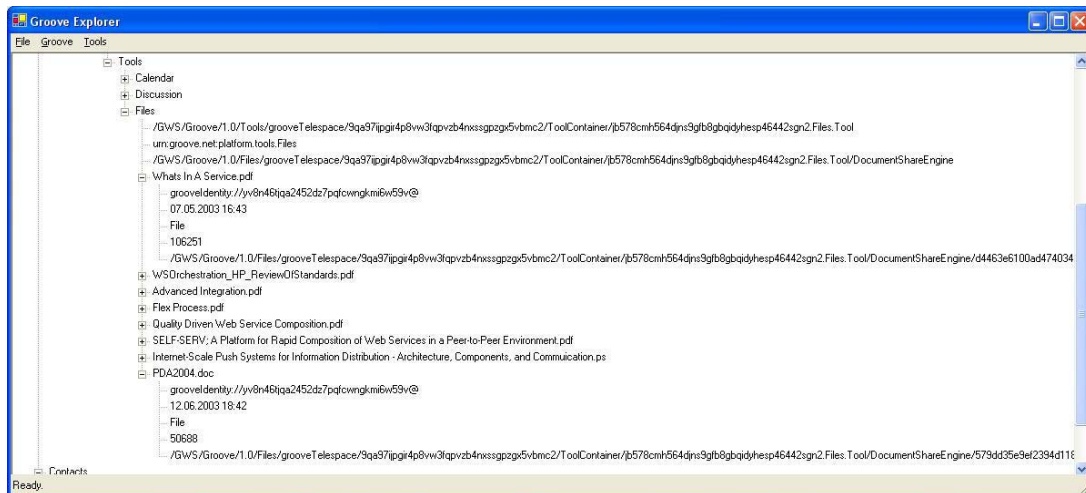

Figure 6. Groove Explorer for Services

Groove Web services provide a mechanism for SOAP clients to register as listeners for Groove events and check an event queue for messages. The SOAP clients receive events that are generated by changes in the underlying data. These changes can be initiated by the Groove user on client system providing the Web services, by another Groove user in a shared space, or by the SOAP client itself. Table 1 lists the events provided by Groove Web services provided by Groove 2.5:

Table 1. Groove Web services

Service	Event Class	Event Types
GrooveCalendar	urn:groove-net:CalendarEvent	CalendarAddEventData, CalendarDeleteEventData, CalendarUpdateEventData
GrooveContacts	urn:groove-net:ContactEvent	ContactAddEventData, ContactDeleteEventData, ContactUpdateEventData
GrooveDiscussion	urn:groove-net:DiscussionEvent	DiscussionAddEventData, DiscussionDeleteEventData, DiscussionUpdateEventData
GrooveFilesBase64 GrooveFilesDIME	urn:groove-net:FileEvent	FileAddEventData, FileDeleteEventData, FileRenameEventData, FileUpdateEventData
GrooveSpaces	urn:groove-net:SpaceEvent	SpaceAddEventData, SpaceDeleteEventData, SpaceRenameEventData
GrooveTools	urn:groove-net:ToolEvent	ToolAddEventData, ToolDeleteEventData, ToolRenameEventData

3.2 Use case "paper review"

A paper review process is used to demonstrate the possibilities of Groove Workspace and its Web service interface. Therefore a new shared space is created and all reviewers are invited by a review coordinator. The new shared space includes at least the Groove tools "Files" to make all papers available for the review process and "Discussion" to allow a minimum of interaction between the reviewers and the coordinator (see Figure 4). Groove Workspace would also provide a special "Document Review" tool, but as it is not accessible during a Web service interface, it could not be used by third party software and is restricted to run Groove Workspace for every reviewer.

The used shared space "PDP 2004" in Figure 4 shows the Groove tool "Files" including several papers to review, and an empty directory for each reviewer (Dustdar, Gall, and Schmidt) which will include the reviewed papers at the end of the review process. As creator of "PDA 2004", Roman Schmidt also acts as review coordinator. The reviewers Dustdar and Gall are running third party software tools (e.g. Colligo for PDAs [9] or Caramba [2]) which will interact with Groove by their Web service interface (therefore they are not visible to the coordinator by the Groove Workspace). The only possibility for communication between all reviewers and the coordinators is the Groove tool "Discussion", which allows creating topics, writing messages, and writing responses for all participants.

To stay up-to-date, all reviewers using the Web service interface have to subscribe for events possibly raised by the used tools (see Figure 7). Dustdar and Gall subscribe the Event Class "urn:groove-net:DiscussionEvent" and "urn:groove-net:FileEvent" to receive notifications about new or updated files, or changes in the discussion forum. For example, if Schmidt adds a new file to the Groove tool "Files", all subscribed listeners receive the raised "FileAddEventData" event. As Schmidt uses the Groove Workspace software, it is not necessary to subscribe explicit for events, because it is done automatically.
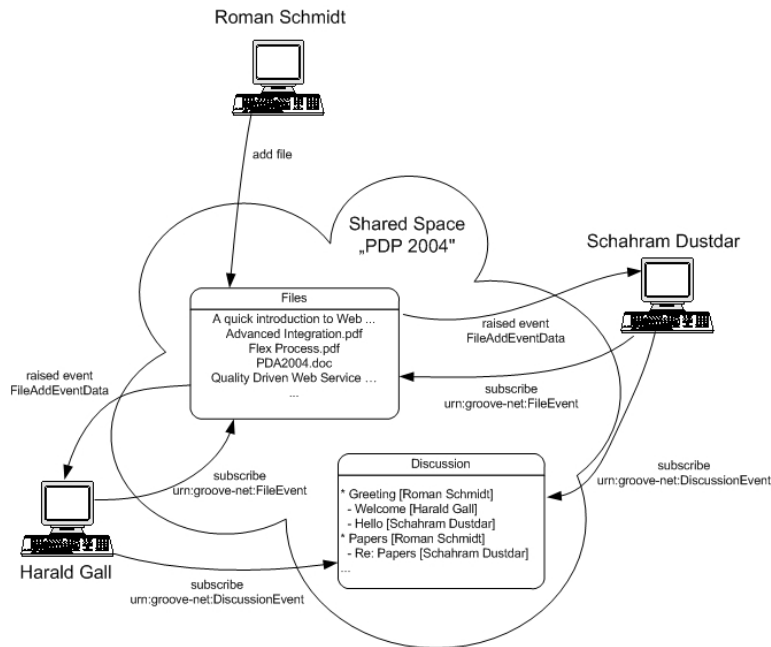
Figure 7. Groove Web service Events

Using subscriptions and events reviewers will be informed about new papers, or updated versions of papers and new messages during a discussion. This allows the coordinator to add new papers and assign these papers to reviewers by adding a new topic to the discussion. Therefore the paper is accessible for the assigned (of course, also for all other participants) and can be reviewed. Afterwards the reviewed version is added to the "Files" under the reviewer's subdirectory. Again all participants will be informed about the progress and the coordinator can detect the end of the review process.

4   Conclusions and Future Work

This paper presented a novel Web services management and configuration architecture with the aim of integrating various Groupware systems into a coherent and configurable architecture. Furthermore we provided a motivational example and a small proof-of-concept implementation extending Groove workspaces with Web services.

Our future work will focus on extending our previously built Teamwork services platform [1] to provide the presented functionalities for gluing together various Groupware systems using Web services technologies.

References

[1]   Dustdar, S. Gall, H. (2003). Architectural concerns in distributed and mobile collaborative systems. 11th Euromicro Conference on Parallel Distributed and Network based Processing (PDP 2003), February, Genoa, Italy, IEEE Computer Society Press.

[2]   Dustdar, S. (2003). Architecture and Design of an Internet-enabled Integrated Workflow and Groupware system. Business Process Management Journal, Special Issue on Internet-enabled Workflow, (forthcoming).

[3]   http://www.ebxml.org

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

[5]   http://www.groove.net

[6]   www.oasis-open.org

[7]   World Wide Web Consortium, SOAP (Simple Object Access Protocol), http://www.w3.org/TR/2001/WD-soap12-part1-20011002

[8]   World Wide Web Consortium, WSDL (Web-Service Description Language), http://www.w3.org/TR/wsdl

[9]   http://www.colligo.net

[10]   Casati, F., Machiraju, V. (2003) Business Visibility with Web services: Making sense of your IT operations and of what they mean to you. Proceedings of UMICS 2003, collocated with CAiSE 2003, Velden, Austria, pp. 123-135.