Web Infrastructure and Coordination Architectures for Collaborative Applications – Shared Artifacts, a Shared Language, or Shared Spaces?[*]

edited by
Gabriele Kotsis[†] *gabi@ani.univie.ac.at*
Gustaf Neumann[‡] *neumann@wi-inf.uni-essen.de*

Abstract

The attendees of the workshop "Web Infrastructure and Coordination Architectures for Collaborative Applications" tried to develop a joint opinion how the Web and its infrastructure should develop to become a productive environment for collaborative applications. This paper sketches the different approaches presented in the workshop and summarizes the consolidated results of the discussions.

1 Introduction

The workshop "*Web Infrastructure and Coordination Architectures for Collaborative Applications"* at WETICE'99 brought together attendees from academia and industry. Half of the participants came from Europa, the other half was from the US. After the referee process 10 papers were selected for presentation and discussion at the workshop. The papers can be roughly divided into technical infrastructure papers, shared presence papers and application papers. A final discussion session and the collaborative effort of all participants to produce a summary of the workshop results rounded off the sessions.

This workshop continues two threads of workshops in the WETICE series that were held over the last three years. In general these workshops addressed the question how Web techniques can be used to achieve or to improve collaboration within or between organizations, and which coordination mechanisms could be used in such an architecture.

It was our goal to address both, technical and organizational issues. The involved organizations are typically enterprises in different spheres of power such that it is not reasonable to impose a common structure, or to align the structure for a certain collaboration purpose. But how can a system enable the participants for flexible and effective collaboration?

This question lead us to this year's workshop theme of "*Shared Artifacts, a Shared Language, or Shared Spaces"*:

- Should we concentrate on developing an infrastructure for sharing artifacts (documents, files, data)? Shared artifacts require a high degree of coordination: the participants have to agree about the tools manipulating the artifacts, and about their structures and identifiers of the artifacts. Often these structures of artifacts are well established within a sphere of control and therefore hard to change for the collaboration with external entities.

- Therefore an alternative approach is to develop a shared language to exchange business information which represents an external representation of the private artifacts. Various approaches based of XML or RDF look very promising for providing a scalable, canonical and standardized language which is still flexible enough to cope with changing business requirements.

- The shared language mimics shared artifacts to a certain degree. For which kind of collaborations do we need a richer communication infrastructure supporting a shared spaces, where people are "in the system" rather than purely using artifacts of a system? A shared space implies the notion to be at a certain location in the systems, other participants are aware of the presence of collaborators.

The call for papers focused on the main question whether current Web techniques can serve as an infrastructure for both developing and implementing networked, collaborative business applications in a (global) distributed business environment. While the current Web is certainly not a perfect environment for collaboration, we think it is powerful enough not to inhibit effective collaboration.

[*] Published in: Proceedings of WETICE 99, IEEE 8th Intl. Workshops on Enabling Technologies, Stanford, CA, June, 1999.

[†] Universität Wien, Angewandte Informatik, Austria

[‡] Universität Essen, Wirtschaftsinformatik, Germany

We tried to position this workshop not only to address technical issues of “collaborating systems” but of “collaboration systems” as well, that is, to focus on infrastructures for enabling humans to collaborate. The workshop addressed issues such as Web based systems for knowledge management and sharing, in their architecture, technical implementation, in coordination languages and their integration in everyday’s work.

During the discussions Christopher Landauer developed the hypothesis that rather than addressing the workshop theme from the three different points of view of the subtitle, we can try to concentrate on the issue of “shared services”. Break the triversity! Actually, it worked out very well: the papers could be easily positioned to provide services at various levels in the ideal “collaboration system”.

2 Shared Services

What are these services, how can we specify these services, how do these services work together? These are tough questions for which we have no definite answer yet. The success of the Web is based on the low entry cost for accessing and publishing information, which led us to asynchronous collaboration through shared artifacts. We hope that a Web of services can provide the same low entry costs of using existing and providing new services based on each other. These shared services should allow automated processing steps leading to a value chain of services that will lead us to a much richer Web-based computing environment, where mere humans are able to contribute on their level of expertise (e.g. infrastructural or application specific) with reasonable effort.

In the following we are trying to identify the services that are needed most for collaborative applications. Basic services for collaboration include the coordination of activities and the exchange of information.

However, managing and sharing information requires efficient mechanisms being able to cope with consistency while at the same time maintaining a satisfying performance of the system. H. Yu, D. Estrin, and R. Govindan have addressed the issue of scalability of **information dissemination services** in *A Hierarchical Proxy Architecture for Internet-scale Event Services*. They propose a tree based hierarchy glued together by multicast groups. Proxies within a group exchange periodic heartbeat messages to indicate liveness. Registrations of and subscriptions to information sources are propagated through the hierarchy only where and when needed to avoid flooding of messages. Further improvements aim at reducing the latency within the proxy hierarchy by providing “tree shortcuts”.

In *A Coordination Service for Distributed Applications* by Singh and Gopalan, an object-oriented framework is proposed, following the idea of separating the different aspects of (collaborative) application development, namely session management, coordination, and information exchange. A **coordination specification service** allows for a description of the relations among the participants’ interactions in terms of a Petri net. The **information exchange service** provides mechanisms to exchange and save data among entities participating in the collaboration. The **session management service** allows users to create sessions, locate existing sessions and enforce rules on sessions at run time.

Chris Landauer and Kirstie Bellman gave their point of view on integration of services in *Virtual Webworlds: Extending the Web for Collaboration*. Their vision is the integration of the concept of shared artifacts which is present in the current WWW and the concept of shared presence as it is known from Multi-User virtual Environments (MUVEs), coming up with a proposal for Virtual Webworlds providing **integration services**.

Another aspect of integration, namely integration of different types of media, was addressed by Michelle Potts Steves, Wo Chang, and Amy Knutilla in *Supporting Manufacturing Process Analysis and Trouble Shooting with ACTS*. They have developed a framework for cooperation of remote and local experts in a manufacturing process. Focus was given to the provision of services for temporal and spatial synchronization and for annotation of different kinds of information on the manufacturing process for remote problem detection and solving. A SMIL based prototype implementation is available providing these **synchronization services**.

Michael Berger was discussing the problem on how to re-synchronize phases of asynchronous collaborative work by means of history-based **re-integration services**. In his paper, *History-Based Re-integration of Replicated Shared Workspaces* a set of basic services to merge and resolve conflicts are described. A demonstration implementation of the service components is available.

At a higher level, services are needed supporting the actual functions in a cooperation or providing awareness information to the users.

The NESSIE environment, presented by Eckhard Meier, Wolfgang Prinz, and Wolfgang Broll in their paper on *Augmenting Cooperative Settings by Shared Awareness Spaces*, provides **awareness services** at various levels. Being application independent, NESSIE supports the capturing and transmission of sensory input on user’s social and task oriented activities and their transformation into (nearly) any kind awareness information. NESSIE has been implemented and tested at the GMD-FIT research group.

TeamVote, presented by Alois Ferscha and Christoph Scheiner in *Collective Choice in Virtual Teams* is a web-based tool providing services for group decision making. Recalling a known results from literature, namely that there exists no optimum voting procedure, the authors have provided a set of **collective choice services** allowing the easy set up of an electronic, web-based offering the choice among several voting procedures. A prototype of the system has been developed and can be obtained from the authors.

Giving the end user's point of view, Sarah Drummond and Cornelia Boldyreff reported their experiences with *SEGWorld: A WWW-based infrastructure to Support the Development of Shared Software Engineering Artifacts*. This BSCW-environment was used at the Department of Computer Science, University of Durham, in a student‘s software engineering project to support collaboration among students. The main findings of the study where that it is crucial to teach students not only how to use the tool but also to teach them the concept of collaboration and sharing of information in a collaborative environment.

Finally, architectural issues were addressed in the remaining two papers.

Lukasz Beca, Geoffrey C. Fox, and Marek Podgorny proposed a *Component Architecture for Building Web-based Synchronous Collaboration Systems*. Following the main principles of component based software development, namely hiding complexity and supporting reusability, sophisticated collaborative applications can be created in a comparatively easy way. The current implementation of the systems provides services for the development of synchronous collaborative JAVA applications or applets based on Java Beans; extensions to other programming languages (C++) are under development.

Fredj Dridi and Gustaf Neumann have proven in their paper on *How to Implement Web-based Groupware Systems based on WebDAV* that an extension of the HTTP protocol, WebDAV, provides a much more comfortable way to develop collaborative applications on the Web by accessing **protocol level services** (normally not accessible by a browser) rather than the extensive use of scripts and servlets or applets. As a demonstration case study, a project repository, where (hypermedia) documents can be shared collaboratively in a controlled way, has been implemented based on WebDAV.

3 Future Directions

Certainly, this collection of papers was not written with the mind to provide shared services. But maybe these ideas can help to trigger a slightly different and new point of view on the area of collaboration and collaborative systems in the Web. During the workshop the following ideas about future developments that might lead towards a Web of services were discussed:

- The development of shared artifacts will continue, but active components will become more important, esp. when they provide a smooth integration into the existing Web.
- Shared languages will continue to develop: Heterogeneity will be required to provide smooth integration and to allow new developments. No single programming language will be sufficient, no single vendor-specific tool is likely to be accepted, no data format will be the perfect one in every situation, etc. Heterogeneity requires open standards for a component-oriented development.
- In order to achieve a higher degree of automated reuse of Web resources (e.g. services) more explicit semantic information will be needed. This can lead to better search tools, robots, agents, mediator programs advertising and publishing their services to others etc.
- The Web of services should provide a layered open architecture where new services can be plugged in with little effort and be used (and reused) by others. The Web of services should not be an “operating system” but an “operations system” that improves (or enables) collaboration between users in their business tasks.
- From the tools we would like to see the development from the rather monolithic browser plus server architecture of today towards a distributed micro-kernel-like architecture for services, that is controlled by an lightweight instrument, an *xPURC*$^{\text{TM}}$, an extensible, programmable, universal remote control.

We would be glad if new research could be triggered by these ideas of our workshop.

The slides containing the final presentation of the group discussions that were produced at the workshop are available on the Web at the location `http://nestroy.wi-inf.uni-essen.de/workshops/WETICE99/slides/`.