FernUniversität in Hagen

Department of Mathematics
and Compute Science

Towards a context-adaptive integrated shared work environment

Stephan Lukosch, Dirk Veiel, Jörg M. Haake

2008

Research Report 1/2008
ISSN 1865-3944

Stephan Lukosch, Dirk Veiel, Jörg M. Haake
Lehrgebiet für Verteilte Systeme für kooperative Arbeits-/Lernumgebungen

Editor:	Dean of the Department of Mathematics and Computer Science
Type and Print:	FernUniversität in Hagen
Distribution:	http://deposit.fernuni-hagen.de/view/departments/miresearchreports.html

Towards a context-adaptive integrated shared work environment

Stephan Lukosch, Dirk Veiel, and Jörg M. Haake
FernUniversität in Hagen
Department for Mathematics and Computer Science
58084 Hagen
Germany
{stephan.lukosch, dirk.veiel, joerg.haake}@fernuni-hagen.de

Abstract

Context-based adaptation can be used to support teams with shared workspace environments best meeting their needs. We propose an ontology representing context in a shared workspace environment, and a conceptual architecture for context sensing, reasoning, and adaptation. First experiences demonstrate the applicability of our approach.

Introduction

Knowledge workers solving complex problems in distributed teams often need many tools (IT systems). This leads to knowledge and information dispersed over different team members, artifacts and systems. Teamwork also requires communication over different systems and artifacts, and leads to difficult communication and coordination. These problems make distributed collaboration difficult.

Current approaches to support distributed collaboration include synchronous and asynchronous multimedia communication systems, repositories for shared documents and shared workspaces systems, notification systems, shared editors, shared calendars, and workflow systems. These systems either leave the organization of collaborative work to the users (e.g., communication systems, repositories) or ignore the changing needs of users to include different tools and artifacts (e.g., notification systems, shared editors, shared calendars, and workflow systems). While shared workspace systems try to combine repositories with coordination and communication support, they still require team members to manually maintain the organization of social processes, artifacts, and tools. This includes the integration of the different tools. As a consequence, teams may fail to adapt their shared workspace to best meet their current needs.

Our idea is to address these problems through a context-adaptive integrated shared work environment that connects the different tools of the team members, and adapts the behavior of the shared work environment according to the group context. We suggest that the overhead for manual adaptation may be decreased by computer-supported adaptation of shared workspace features. We assume that such adaptation may positively impact team performance (e.g., by changing affordances in a way that improves interaction, process and/or output quality).

We propose that such adaptation can be based on context information (i.e. information captured by the system about individual use and group use, and preferences) and on adaptation rules defining in which actual context state to do what kind of adaptation. Adaptation includes modification of the set of applications/services for each user, changing their respective UI, and changing their content. The actual adaptation rules are not static and predefined but need to evolve over time, thus requiring means for users to understand ongoing adaptations and how they can change them. In this way, we see the (evolving) rules as another part of the context information.

Section 2 reviews related work. Section 3 presents our approach. Section 4 reports our experiences. Section 5 concludes the paper and raises questions for future work.

State of the Art

BSCW [1] and CURE [2, 3] are web-based shared work environments offering a variety of collaboration services, e.g. document sharing and communication. CHIPS [4] is a cooperative hypermedia system with integrated process support. TeamSpace [5] offers support for virtual meetings and integrates synchronous and asynchronous team interaction into a task-oriented collaboration space. BRIDGE [6] is a collaboratory that focuses on supporting creative processes and as such integrates a variety of collaboration services. All of the above examples focus on a specific application domain. Though they all offer a variety of services, the systems are independent of each other and do not allow to integrate additional services. Some of them, e.g. CHIPS or CURE are highly tailorable, but they do not adapt themselves to better support collaboration.

The most prominent examples for context-based adaptation focus on single users and consider location as most relevant context information (e.g. [7], [8], [9]) or focus on learner profiles (cf. ITS). Compared to single-user ITS systems, COLER [10] provides a software coach for improving collaboration. CoBrA [11] is an agent-based architecture that uses shared context knowledge represented as an ontology to adapt service agents according to a user's context. The Semantic Workspace Organizer (SWO) [12] is an extension of BSCW. It analyzes user activities and textual documents inside the shared workspace to suggest appropriate locations for new document upload and for document search. The ECOSPACE project aims at providing an integrated collaborative work environment [13, 14]. For that purpose, ECOSPACE uses a service-oriented architecture and provides a series of collaboration services for orchestration and choreography. The orchestration and choreography is based on a ontology which still has to be described [14, 15].

The above approaches focus on domain-specific adaptations. Adaptation based on group context and for multiple users of a cooperative system is intended only by ECOSPACE. The required context information is still an open issue. Similarly, only ECOSPACE supports the integration of different collaboration services within the same shared work environment. In summary, current approaches do not sufficiently support a context-based adaptation of shared work environments.

Approach

We distinguish two parts of a context-adaptive integrated shared work environment: (1) explicit representation of context and (2) conceptual architecture for context sensing, reasoning, and adaptation in such an environment.

Context Definition and Representation

Dey et al. define context as any information used to characterize a situation of an entity where entities may be any object, person or place providing information about the interaction between a user and an application [16]. With this definition, any information may help characterizing the situation of the interaction's participants because it is part of the context itself. For our purposes, we can narrow this definition so that context includes all information which is necessary or helpful to adapt a cooperative workspace to better fit the needs of a collaborating team. This implies that the context contains information about the team as well as about the current collaborative situation.

There exist different approaches to model the context. These approaches range from simple key-value models over graphical models up to sophisticated ontology based models which support validation and reasoning [17]. Similar to CoBrA [11], we need reasoning mechanisms and thus model our context as an ontology using OWL. We use the top-level concepts shown in Figure 1 to describe the context in a shared work environment. Note: Ontology concepts are set in small caps.

Figure 1 Context model for shared workspaces

According our ontology, users collaborate in a GLOBAL COLLABORATION SPACE. This GLOBAL COLLABORATION SPACE consists of LOCAL WORKSPACES which themselves consist of COLLABORATIVE APPLICATIONS. Each COLLABORATIVE APPLICATION is composed from a variety of SERVICES, which access or modify the SHARED MODEL. The SERVICES are bundled in different service classes, which range from DOCUMENT MANAGEMENT up to CONTEXT KNOWLEDGE MANAGEMENT. Each service class further defines basic services, which have to be offered. The SHARED MODEL consists of different ARTIFACTS. A TEAM consists of USERS which perform different ACTIONS using the available SERVICES. The knowledge about the history of actions and the services available for each USER allow us to perform context-sensitive adaptations. These adaptations are represented as ADAPTATION RULES. We will now introduce our conceptual system architecture followed by an example of adaptation.

Conceptual Architecture

Due to the openness of the real world, we may either have to deal with evolving context dimensions or live with a closed set of context dimensions. In both cases new questions arise. How would users be able to deal with evolving context dimensions? How can we enable users to build social solutions around a system with fixed context dimensions? Though our current prototype uses a closed set of context dimensions, our conceptual architecture allows integrating services for extending context dimensions.

Adaptation rules need to evolve over time, thus requiring means for users to understand ongoing adaptations (traceability, reflection, understandability of changes) and how they can change them. Again, this is a social process, which we want to base on negotiations within the team. Of course, social practice may become best practice in the organization and thus may need to be shared.

Considering these observations, our conceptual architecture consists of four layers: *Application UIs*, *Adaptation Server*, *Collaboration Services*, and *Shared Model* (cf. Figure 2).

A flexible adaptive system executes a cycle of (1) user interaction, (2) sensing user activities, (3) adapting system behavior, and (4) modifying adaptation knowledge (e.g., if users want to change adaptation rules).

We use the sensing functionality to illustrate the interactions between the different components of our architecture. The UI-part of a collaborative application (*Application UIs*) is used to collect relevant information about the interaction between the user and the application. This information (including the service calls) is forwarded to the *Sensing* component of the *Adaptation Server,* which extracts relevant information and updates *Context Knowledge* via the *Context Knowledge Service Provider*. This information is passed to the *Collaborative Application Manager (CAM)*, which triggers the services of the corresponding *Collaborative Application*. Applications can use several basic services from different *Service Providers* to implement the application logic. Thereby, we can integrate different services into an application and adapt application behavior across different service providers. Furthermore, new services can be added, possibly requiring an extension of the service concept in our ontology.

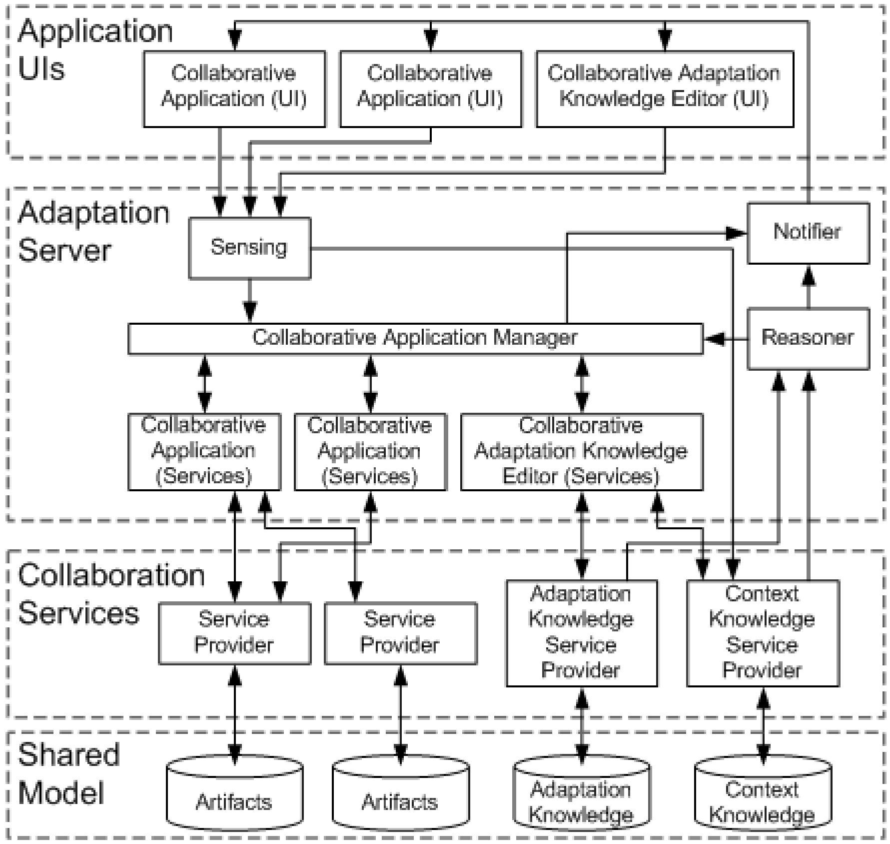

Figure 2 Conceptual Architecture

Next, we describe the adaptation functionality. The *Reasoner* uses the current *Adaptation* and *Context Knowledge* via the *Adaptation* and *Context Knowledge Service Providers* to find relevant adaptation rules that modify the set of applications/services for each user, change their respective UI, and/or change their content. The information about the modifications/changes is passed to the CAM to update the current service configuration. The *Notifier* sends the adaptation information to the subscribed UI-parts of a *Collaborative Application*. The UI-parts then use the current service configuration in the CAM to refresh the view.

The following example illustrates our adaptation approach. Consider two scientists that collaboratively write a paper. When both of them edit the same section of the document at the same time, a potential for collaboration or conflict occurs. One possible way to use such a potential is to establish a communication channel between both users. The corresponding adaption rule looks like this:

```
results = reasoner.query(sparql-query);
if (results > 1) then {
  cc = comm.createChannel(results);
  notifier.notifyApplication(cc); }
```

The SPARQL query necessary to recognize such a situation searches for users that edit the same ARTIFACT and are capable of using the SYNCHRONOUS COMMUNICATION service CHAT. The query is defined as follows:

```
SELECT DISTINCT ?user1 ?service
WHERE {
  ?edit rdf:type :Edit.
  ?service rdf:type :Chat.
  ?user1 :performs ?edit.
  ?user2 :performs ?edit.
  ?user1 :isCapableOf ?service.
  ?user2 :isCapableOf ?service.
  ?edit :modifies ?artifact. }
```

The *Collaborative Adaptation Knowledge Editor (CAKE)* is used to support the modification of adaptation knowledge. It allows users to access, review, edit, and create new user-defined adaptation rules. CAKE is a collaborative application using our architecture meaning that it can be adapted as well. CAKE uses the *Adaptation* and *Context Knowledge Service Providers* to access and modify the corresponding adaptation information.

Implementation

The current prototype manages our ontology, i.e. our context, using the semantic web framework Jena[1]. The *Adaptation Server* is based on Equinox[2] and realizes all components as so-called bundles in OSGi. In the prototype, we integrated two *Collaborative Applications*. First, we extended CURE [2, 3] to provide adapters for the service classes ACCESS RIGHT MANAGEMENT, ASYNCHRONOUS AWARENESS, ASYNCHRONOUS COMMUNICATION, DOCUMENT MANAGEMENT, USER and WORKSPACE MANAGEMENT. Second, we integrated an Openfire[3] server through adapters for the service classes SYNCHRONOUS AWARENESS and SYNCHRONOUS COMMUNICATION in our ontology. The UI-parts for these service classes are implemented as plug-ins for Eclipse[4] based on the generic interfaces defined for each service class in our ontology.

Experiences

We split up the evaluation in three phases. In the first phase, we tested our approach by implementing the conceptual architecture, integrating two collaborative applications, and conducting functional tests. Currently, we are integrating further applications to test the coverage of service classes from our ontology.

---

[1] http://jena.sourceforge.net/

[2] http://www.eclipse.org/equinox/

[3] http://www.igniterealtime.org/projects/openfire/

[4] http://www.eclipse.org/

In the second phase, we have setup a test environment and conducted expert walkthroughs in different work scenarios, e.g. collaborative planning or writing. Feedback from these experts indicates both, the usefulness as well as problems caused by the adaptation in general and of specific rules.

In the third phase, we will evaluate our architecture with the integrated collaborative applications in real-world settings which are based on knowledge-intense processes, such as planning games in pair programming or scientific paper production. From these evaluations, we expect further feedback on the existing adaptation rules. We also want to use the third phase to identify best practices for collaboration leading to further adaptation rules.

Conclusion

In this paper, we introduced an explicit representation of context and a conceptual architecture for context sensing, reasoning, and adaptation in an integrated shared work environment. We integrated first collaborative applications to show the feasibility of our approach.

Our approach exceeds current approaches by defining an ontology-based context model for shared workspaces and an architecture which allows to integrate a variety of services to adapt the shared workspace to best meet the current needs of collaborating users.

First experiences show the usefulness as well as problems caused by the adaptation. This highlights the importance of user-defined adaptation rules as well as process support for negotiating adaptations and adaptation rules.

References

1. Appelt, W. and P. Mambrey. *Experiences with the Bscw Shared Workspace System as the Backbone of a Virtual Learning Environment for Students*. in *Proceedings of ED-MEDIA99*. 1999.
2. Haake, J.M., T. Schümmer, A. Haake, M. Bourimi, and B. Landgraf. *Supporting Flexible Collaborative Distance Learning in the Cure Platform*. in *Proceedings of the Hawaii International Conference On System Sciences (HICSS-37)*. 2004.
3. Haake, J.M., A. Haake, T. Schümmer, M. Bourimi, and B. Landgraf. *End-User Controlled Group Formation and Access Rights Management in a Shared Workspace System*. in *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 2004. p. 554-563.
4. Wang, W. and J.M. Haake, *Tailoring Groupware: The Cooperative Hypermedia Approach.* Computer Supported Cooperative Work, 2000. **9**(1): p. 123-146.
5. Geyer, W., H. Richter, L. Fuchs, T. Frauenhofer, S. Daijavad, and S. Poltrock. *A Team Collaboration Space Supporting Capture and Access of Virtual Meetings*. in *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*. 2001. p. 188-196.
6. Farooq, U., J.M. Carroll, and C.H. Ganoe. *Supporting Creativity in Distributed Scientific Communities*. in *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*. 2005. p. 217-226.
7. Schilit, B., N. Adams, and R. Want. *Context-Aware Computing Applications*. in *First Annual Workshop on Mobile Computing Systems and Applications (WMCSA)*. 1994.

8. Abowd, G.D., C.G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, *Cyberguide: A Mobile Context-Aware Tour Guide.* Wireless Networks, 1997. **3**(5): p. 421-433.
9. Kindberg, T., J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic, *People, Places, Things: Web Presence for the Real World.* Mob. Netw. Appl., 2002. **7**(5): p. 365-376.
10. Constantino-González, M.d.l.A. and D.D. Suthers, *Automated Coaching of Collaboration Based on Workspace Analysis: Evaluation and Implications for Future Learning Environments*, in *Proceedings of the 36th Hawai`i International Conference on the System Sciences (HICSS-36).* 2003, IEEE Press.
11. Chen, H., T.W. Finin, and A. Joshi. *Semantic Web in the Context Broker Architecture*. in *PerCom*. 2004. p. 277-286.
12. Prinz, W. and B. Zaman. *Proactive Support for the Organization of Shared Workspaces Using Activity Patterns and Content Analysis*. in *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*. 2005. p. 246-255.
13. Prinz, W., H. Loh, M. Pallot, H. Schaffers, A. Skarmeta, and S. Decker. *Ecospace - Towards an Integrated Collaboration Space for Eprofessionals*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. 2006. p. 39-45.
14. Martínez-Carreras, M.A., A. Ruiz-Martínez, F. Gómez-Skarmeta, and W. Prinz. *Designing a Generic Collaborative Working Environment*. in *IEEE International Conference on Web Services (ICWS 2007)*. 2007. p. 1080-1087.
15. Vonrueden, M. and W. Prinz. *Distributed Document Contexts in Cooperation Systems*. in *Modeling and Using Context, 6th International and Interdisciplinary Conference, CONTEXT 2007*. 2007. p. 507-516.
16. Dey, A.K., G.D. Abowd, and D. Salber, *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications.* Human-Computer Interaction, 2001. **16**(2, 3, & 4): p. 97-166.
17. Strang, T. and C. Linnhoff-Popien, *A Context Modeling Survey*, in *First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*. 2004: Nottingham, England.