

Kontextbasierte Adaption kooperativer Arbeitsbereiche

Dirk Veiel, Joerg M. Haake, Stephan Lukosch

Fakultät für Mathematik und Informatik
FernUniversität in Hagen
Universitätsstr. 1
58084 Hagen

{dirk.veiel, joerg.haake, stephan.lukosch}@fernuni-hagen.de

Abstract: Die kontextbasierte Adaption von gemeinsamen Arbeitsbereichen kann helfen, ihr Verhalten an die wechselnden Bedürfnisse von Teams anzupassen. Auf Basis einer OWL-basierten Repräsentation des Gruppenkontexts und einer flexiblen, dienstbasierten Architektur können Adaptionsregeln zur Anpassung des kooperativen Arbeitsbereichs formuliert und ausgeführt werden. Erste Tests belegen die Realisierbarkeit des Ansatzes.

1 Einleitung

Kooperative Arbeitsbereiche [AM99, HSH+04, WH00, GRF+01, FCG05] unterstützen verteilte Wissensarbeiter bei der Problemlösung durch Bereitstellung gemeinsamer Daten, deren gemeinsamer Manipulation sowie durch Kommunikations- und Koordinationswerkzeuge. Je nach Problemstellung, Phase und Eigenart des Teams werden unterschiedliche Sichten, Werkzeuge und Funktionalitäten benötigt, um die Kooperation optimal zu unterstützen. Heutige kooperative Arbeitsbereiche müssen von den Benutzern manuell an ihre Bedürfnisse angepasst werden. Aufgrund der Komplexität und des notwendigen Aufwands unterbleiben solche Anpassungen häufig. Die Teams werden deshalb nicht optimal unterstützt und erleiden Leistungsverluste.

Wir stellen eine Architektur eines kontextadaptiven, integrierten, gemeinsamen Arbeitsbereichs vor, der die Artefakte und Werkzeuge der Teammitglieder verbindet und das Verhalten des gemeinsamen Arbeitsbereichs an den Gruppenkontext adaptiert. Dazu nutzen wir eine Repräsentation des Kontexts und Adaptionsregeln. Adaption kann hierbei sowohl die Modifikation der Anwendungen bzw. Dienste jedes Nutzers, die Änderung der entsprechenden Oberfläche als auch deren Inhalte bedeuten. Wir werden zeigen, dass sich der Mehraufwand, der sich durch die manuelle Adaption ergibt, durch eine computerunterstützte Adaption verringern lässt.

Wir stellen zuerst unsere Kontextrepräsentation und die konzeptuelle Architektur für kontextbasierte Adaption vor, bevor wir auf die Implementierung und Nutzungserfahrungen eingehen. Zum Abschluss vergleichen wir unseren Ansatz mit verwandten Arbeiten.

2 Kontextdefinition und -repräsentation

Dey et al. definiert Kontext als „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“ [DAS01]. Diese Definition verdeutlicht, dass jede Information über die Interaktion und die entsprechenden Beteiligten zum besseren Verständnis und zur Beschreibung der aktuellen Situation und damit auch des Kontexts beitragen kann. Wir schränken diese Definition insofern ein, dass wir alle Informationen in den Kontext aufnehmen, die notwendig oder hilfreich sind, um einen kooperativen Arbeitsbereich an die Bedürfnisse eines kooperierenden Teams adaptieren zu können. Daraus ergibt sich, dass der Kontext sowohl Informationen über das Team als auch über die aktuelle Kooperationssituation enthält.

Es gibt diverse Möglichkeiten den Kontext zu modellieren. Die Ansätze reichen hierbei von einfachen Schlüssel-Wert-Paaren über grafische Modelle bis hin zu komplexen Ontologie-basierten Modellen, die eine Validierung und Reasoning unterstützen [SLP04]. Ähnlich wie bei CoBrA [CFJ04], benötigen wir die Möglichkeit, Reasoning zu betreiben und modellieren deshalb unseren Kontext mit Hilfe von OWL (<http://www.w3.org/2004/OWL/>).

Gemäß unserer Ontologie kollaborieren USER in einem GLOBAL COLLABORATION SPACE, der sich wiederum aus LOCAL WORKSPACES zusammensetzt (Anmerkung: Ontologiekonzepte sind als KAPITÄLCHEN geschrieben.). Diese bestehen ihrerseits aus COLLABORATIVE APPLICATIONS. Jede COLLABORATIVE APPLICATION nutzt eine Menge von SERVICES, die auf das SHARED MODEL zugreifen und es modifizieren können. Die SERVICES sind sogenannten Serviceklassen zugeordnet, die vom DOCUMENT MANGEMENT bis hin zum CONTEXT KNOWLEDGE MANGEMENT reichen. Jede Serviceklasse definiert ihrerseits wiederum Basisdienste, die angeboten werden müssen. Das SHARED MODEL besteht aus verschiedenen ARTIFACTS. Ein TEAM besteht aus USER, die diverse ACTIONS ausführen, die SERVICES nutzen. Kontextsensitive Adaption wird durch das Wissen über die Historie der ACTIONS und die Dienste, die jeder USER nutzen kann, ermöglicht. Diese Adaptionen werden über die ADAPTATION RULES dargestellt. Nachfolgend stellen wir unsere konzeptionelle Architektur gefolgt von einer beispielhaften Adaption vor.

3 Konzeptuelle Architektur für kontextbasierte Adaption

Ein flexibles, adaptives System durchläuft eine Schleife von (1) Nutzerinteraktion, (2) Ermittlung der Nutzeraktivitäten durch das System (*Sensing*), (3) Adaption des Systemverhaltens und (4) Anpassung des Adaptionswissens (z.B. Änderung einer Adaptionsregel). Unsere konzeptionelle Architektur besteht aus den vier Schichten: *Application UIs*, *Adaptation Server*, *Collaboration Services* und *Shared Model* (vgl. Abbildung 1), in denen diese vier Schritte ablaufen:

Sensing-Funktionalität: Die Bedienoberflächenkomponente einer kollaborativen Anwendung (*Application UIs*) sammelt relevante Informationen über die Interaktion zwischen dem Nutzer und der Anwendung. Diese Informationen (und die zugehörigen Service-Aufrufe) werden an die *Sensing* Komponente des *Adaptation Servers* übergeben, die die für den Kontext relevanten Informationen extrahiert und über den *Context Knowledge Service Provider* in das *Context Knowledge* integrieren lässt. Die Informationen werden anschließend an den *Collaborative Application Manager (CAM)* übergeben, der die Dienste der entsprechenden *Collaborative Application* aufruft. Die Anwendungen können dabei diverse Dienste unterschiedlicher *Service Provider* nutzen, um die benötigte Anwendungslogik zu implementieren. Dadurch können wir verschiedene Dienste in eine Anwendung integrieren und das Verhalten einer Anwendung über verschiedene Dienste hinweg adaptieren. Darüberhinaus können neue Dienste aufgenommen werden, was evtl. eine Erweiterung der Ontologie nach sich zieht.

Adaptionsfunktionalität: Der *CAM* berechnet auf Basis des *ACTION-Logs* im *Context Knowledge* eine Metrik, die die Wahrscheinlichkeit einer Adaption angibt (z.B. basierend auf der Zeit, der Anzahl der *ACTIONS* nach der letzten Adaption oder dem *ACTION-Typ*) und stößt bei Schwellwertüberschreitung den *Reasoner* an. Der *Reasoner* nutzt das aktuelle *Adaptation* und *Context Knowledge* über die entsprechenden *Service Provider*, um relevante Adaptionsregeln zu ermitteln, die die Menge der Anwendungen/Dienste jedes Nutzers modifizieren, die entsprechenden Bedienoberflächen und/oder deren Inhalt ändern. Die Informationen über die Modifikation werden dem *CAM* übermittelt, um die Dienstkongfiguration zu aktualisieren. Die Bedienoberflächen der *Collaborative Applications* erhalten die Informationen über die Adaption vom *Notifier*, um ihre Inhalte gemäß der aktuellen Dienstkongfiguration des *CAMs* zu aktualisieren.

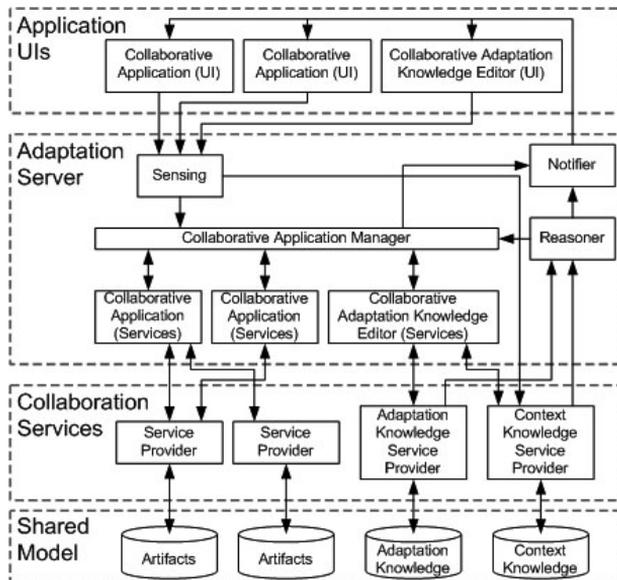


Abbildung 1: Konzeptuelle Architektur

Das folgende Beispiel veranschaulicht unseren Adaptionansatz. Man stelle sich zwei Wissenschaftler vor, die kooperativ einen Artikel schreiben. Wenn beide gleichzeitig denselben Abschnitt des Artikels editieren, entsteht entweder ein Konflikt oder eine Möglichkeit zur Kooperation. Eine Möglichkeit, die beiden Wissenschaftler in diesem Szenario zu unterstützen, wäre, einen Kommunikationskanal zwischen ihnen aufzubauen. Eine entsprechende Adaptionsregel könnte wie folgt aussehen:

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

Die SPARQL-Anfrage (<http://www.w3.org/TR/rdf-sparql-query/>) für das obige Beispiel sucht nach Nutzern, die dasselbe ARTIFACT editieren und aus der Dienstklasse SYNCHRONOUS COMMUNICATION den Dienst CHAT nutzen können:

```
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. }
```

Anpassung des Adaptionwissens: Die initiale Menge von Adaptionsregeln wird entweder vom Nutzer oder vom Entwickler vorgegeben, die dabei den *Collaborative Adaptation Knowledge Editor (CAKE)* verwenden. Die Nachvollziehbarkeit der Regelanwendung und der dadurch bedingten Adaption muss für den Nutzer verständlich sein, sodass er Anpassungen selbst vornehmen kann. *CAKE* wird für die Modifikation des Adaptionwissens benötigt. Er ermöglicht Nutzern, auf Adaptionsregeln zuzugreifen, diese zu überprüfen, zu editieren und neue zu erstellen. Da *CAKE* selbst eine *Collaborative Application* ist kann er ebenfalls adaptiert werden. *CAKE* nutzt den *Adaptation* und *Context Knowledge Service Provider*, um auf die entsprechenden Informationen zuzugreifen und diese ggf. zu modifizieren.

4 Implementation

Der aktuelle Prototyp nutzt das Semantic Web Framework Jena (<http://jena.sourceforge.net/>), um unsere Ontologie, d.h. unseren Kontext, zu verwalten. Der *Adaptation Server* basiert auf Equinox (<http://www.eclipse.org/equinox/>) und bindet alle Komponenten als sogenannte Bundles ein. Für den Prototyp integrierten wir zwei *Collaborative Applications*. Erstens erweiterten wir CURE [HSH+04] um Adapter für die Dienstklassen ACCESS RIGHT MANAGEMENT, ASYNCHRONOUS AWARENESS, ASYNCHRONOUS COMMUNICATION, DOCUMENT MANAGEMENT, USER und WORKSPACE MANAGEMENT. Zweitens entwickelten wir Adapter für die Dienstklassen SYNCHRONOUS AWARENESS und SYNCHRONOUS COMMUNICATION in unserer Ontologie, die auf einen Openfire Server (<http://www.igniterealtime.org/projects/openfire/>) zugreifen. Die Bedienoberflächen für die zuvor genannten Dienstklassen wurden als Plugins für Eclipse (<http://www.eclipse.org/>) entwickelt. Sie nutzen generische Schnittstellen, die für jede Dienstklasse aus unserer Ontologie definiert sind.

5 Erfahrungen

Die Evaluation teilt sich in drei Phasen. In der ersten Phase testeten wir unseren Ansatz, indem die konzeptuelle Architektur implementiert, zwei *Collaborative Applications* integriert und Funktionstests durchgeführt wurden. Derzeit arbeiten wir daran, zusätzliche Anwendungen zu integrieren, um die Abdeckung der Dienstklassen unserer Ontologie zu testen. In der zweiten Phase haben wir eine Testumgebung aufgebaut, in der wir mit Experten zusammen verschiedene Arbeitsabläufe (z.B. kooperatives Planen oder Schreiben) durchliefen. Die Rückmeldungen der Experten bestätigten den Nutzen unseres Ansatzes. In der dritten Phase werden wir unsere Architektur und die darin enthaltenen kollaborativen Anwendungen unter realen Bedingungen evaluieren und verstärkt Rückmeldungen bzgl. der bereits vorhandenen Adaptionen von den Nutzern einfordern und daraus Best Practices für Adaption ermitteln.

5 Diskussion

Wir unterstützen die semi-automatische Anpassung kooperativer Arbeitsbereiche an wechselnde Gruppenbedürfnisse durch eine explizite Repräsentation des Gruppenkontexts auf Basis von OWL und eine konzeptuelle Architektur für kontextbasierte Adaption mit Hilfe von Adaptionenregeln und einer flexiblen Integration von kooperativen Diensten in einen gemeinsamen Arbeitsbereich. Eine prototypische Implementierung und die Integration zweier kooperativer Anwendungen zeigt die Realisierbarkeit unseres Ansatzes.

Heutige kooperative Arbeitsbereiche unterstützen üblicherweise keine Adaption. Kontextadaptive Ansätze konzentrieren sich auf einzelne Nutzer und betrachten die Ortsangaben als wichtigste Kontextinformation (vgl. [SAW94, AAH+97, KBM+02]) oder betrachten Lern(er)profile (ITS). COLER [CGS03] nutzt, im Vergleich zu anderen Einbenutzer-ITSen, einen Softwaretutor, um die Kollaboration zu verbessern. CoBrA [CFJ04] nutzt eine Agenten-basierte Architektur, in der das gemeinsame Kontextwissen als Ontologie vorliegt, um die Serviceagenten an die Bedürfnisse des Nutzerkontexts zu adaptieren. Eine Erweiterung von BSCW ist der Semantic Workspace Organizer (SWO) [PZ05], der die Nutzeraktivitäten und die Textdokumente im gemeinsamen Arbeitsbereich analysiert, um Ablageorte für neue Dokumente und ähnliche Dokumente vorzuschlagen. Das ECOSPACE Projekt will eine integrierte, kollaborative Arbeitsumgebung zur Verfügung stellen [PLP06, MRG07]. Es setzt dabei eine Service-orientierte Architektur ein und bietet eine Vielzahl von Kollaborationsdiensten für die Orchestrierung und Choreografie von Diensten an. Hierbei wird eine Ontologie eingesetzt, die noch nicht genauer beschrieben ist [MRG07, VP07]. Unser Ansatz geht über diese Ansätze hinaus, da wir ein Ontologie-basiertes Kontextmodell für gemeinsame Arbeitsbereiche und eine Architektur anbieten, in die eine Vielzahl von Diensten integriert werden können, um den gemeinsamen Arbeitsbereich bestmöglich an die Bedürfnisse der kooperierenden Nutzer anzupassen.

Erste Erfahrungen zeigen sowohl den Nutzen als auch die Probleme, die von der Adaption ausgehen. Das verdeutlicht, dass es wichtig ist, sowohl nutzerdefinierte Adaptionsregeln als auch eine Prozessunterstützung für die Verhandlung über Adaptionen und Adaptionsregeln zu unterstützen.

Literaturverzeichnis

- [AAH+97] Abowd, G.D., et al., Cyberguide: a mobile context-aware tour guide. *Wireless Networks*, 1997. 3(5): p. 421-433.
- [AM99] 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.
- [CFJ04] Chen, H., T.W. Finin, and A. Joshi. *Semantic Web in the Context Broker Architecture*. in *PerCom*. 2004.
- [CGS03] 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.
- [DAS01] 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.
- [FCG05] Farooq, U., J.M. Carroll, and C.H. Ganoë. Supporting creativity in distributed scientific communities. in *Proceedings of the 2005 ACM SIGGROUP conference on Supporting group work*. 2005. New York, NY, USA: ACM.
- [GRF+01] Geyer, W., et al. A team collaboration space supporting capture and access of virtual meetings. in *Proceedings of the 2001 ACM SIGGROUP Conference on Supporting Group Work*. 2001. New York, NY, USA: ACM.

- [HSH+04] Haake, J.M., et al. Supporting flexible collaborative distance learning in the CURE platform. in Proceedings of the Hawaii International Conference On System Sciences (HICSS-37). 2004: IEEE Press.
- [KBM+02] Kindberg, T., et al., People, places, things: web presence for the real world. Mobile Network Applications, 2002. 7(5): p. 365-376.
- [MRG07] Martínez-Carreras, M.A., et al. Designing a Generic Collaborative Working Environment. in IEEE International Conference on Web Services (ICWS 2007). 2007.
- [PLP06] Prinz, W., et al. ECOSPACE - Towards an Integrated Collaboration Space for eProfessionals. in International Conference on Collaborative Computing: Networking, Applications and Worksharing. 2006.
- [PZ05] Prinz, W. and B. Zaman. Proactive support for the organization of shared workspaces using activity patterns and content analysis. in Proceedings of the 2005 ACM SIGGROUP conference on Supporting group work. 2005: ACM, New York, NY, USA.
- [SAW94] Schilit, B., N. Adams, and R. Want. Context-Aware Computing Applications. in First Annual Workshop on Mobile Computing Systems and Applications (WMCSA). 1994.
- [SLP04] 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.
- [VP07] Vonrueden, M. and W. Prinz. Distributed Document Contexts in Cooperation Systems. in Modeling and Using Context, 6th International and Interdisciplinary Conference, CONTEXT 2007. 2007: Springer-Verlag Berlin Heidelberg.
- [WH00] Wang, W. and J.M. Haake, Tailoring Groupware: The Cooperative Hypermedia Approach. Computer Supported Cooperative Work, 2000. 9(1): p. 123-146.