

Universität Paderborn

Diplomarbeit

Analyse und Spezifikation von Anforderungen an Contextual Collaborative Frameworks und beispielhafte Evaluierung der IBM Workplace Architektur

Prof. Dr. Ludwig Nastansky

Sommersemester 2006

Betreuer: Ingo Erdmann

vorgelegt von:

Iris Fehle mann-Heindörfer

Matrikel Nr.: 3482281

Wirtschaftsinformatik

Fliederweg 7

33100 Paderborn

Iris_Fehle mann@notes.upb.de

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Paderborn, den

INHALTSVERZEICHNIS

1. EINLEITUNG	1
1.1 Motivation der Arbeit	1
1.2 Aufbau und Zielsetzung der Arbeit	2
2. GRUNDLAGEN	4
2.1 Contextual Collaboration als Kollaborationskonzept in CSCW und Groupware	5
2.2 Geschäftsprozesse.....	7
2.3 „Contextual Collaboration“ (kontextuelle Kollaboration) als Begriff.....	10
2.3.1 Der Geschäftsprozess als Kontext der kontextuellen Kollaboration	11
2.3.2 Kontextinformationen und Kontextwissen	13
2.3.3 Historische Entwicklung kollaborativer Systeme.....	14
2.3.4 Rahmenbedingungen für kontextuelle Kollaboration.....	16
2.4 Grundlagen zur Portalarchitektur	20
2.4.1 Basiskomponenten von Portalen	21
2.4.2 Basisdienste von Portalen.....	23
3. ANFORDERUNGEN UND KONZEPTE EINES FRAMEWORKS ZUR KONTEXTUELLEN KOLLABORATION	25
3.1 Anwendungsszenario: Kontextuelle Kollaboration in einem Logistikunternehmen.....	25
3.1.1 Das Szenario.....	25
3.1.2 People-Awareness im Kontext einer CRM-Applikation	26
3.1.3 Senden einer E-Mail aus dem Kontext einer CRM-Applikation	26
3.1.4 Anlegen einer Activity zur ad-hoc Koordination einer Angebotserstellung.....	27
3.1.5 Kontextuelle Integration einer CRM-Applikation und einer Dokumentbibliothek	28
3.1.6 Zufügen eines Dokumentes zum Kontext einer Activity.....	28
3.1.7 Erstellen eines Benachrichtigungsprofils für eine Activity	29
3.1.8 Erstellen eines Teamspace zur Durchführung des Projektes	30
3.1.9 Der Shared-Teamspace als „Composite-Application“.....	30
3.1.10 Kontextuelle Kollaboration im Teamspace	31
3.1.11 Navigation zwischen verschiedenen Kontexten: Projekt, Kunde, Teamspace	32
3.1.12 Kollaboration und Device-based-Awareness.....	33
3.1.13 Gemeinsames Nutzen eines Dokumentes im Kontext des Teamspace.....	33

3.2 Das Vier-Stufenmodell der kontextuellen Kollaboration	35
3.2.1 Stufe 1: Starten kollaborativer Funktionen aus dem Arbeitskontext	36
3.2.1.1 Vorteile	36
3.2.1.2 Anforderungen	37
3.2.1.3 Systemmerkmale.....	38
3.2.2 Stufe 2: Einbinden des Arbeitskontextes in die Kollaboration.....	38
3.2.2.1 Vorteile	39
3.2.2.2 Anforderungen.....	40
3.2.2.3 Systemmerkmale.....	40
3.2.3 Stufe 3: Persistierung der Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen	40
3.2.3.1 Vorteile	43
3.2.3.2 Anforderungen.....	44
3.2.3.3 Systemmerkmale.....	45
3.2.4 Stufe 4: Navigation durch das multidimensionale Geflecht von Kontextbeziehungen	45
3.2.4.1 Vorteile	48
3.2.4.2 Anforderungen.....	48
3.2.4.3 Systemmerkmale.....	49
3.3 Strukturmerkmale eines idealtypischen Frameworks für kontextuelle Kollaboration	50
3.3.1 Kollaborative Funktionalitäten	52
3.3.2 Komponentisierung	54
3.3.3 Komponentenintegration	55
3.3.3.1 Integration auf API-Ebene	56
3.3.3.2 Integration auf Komponentenebene.....	57
3.3.3.3 Integration auf Objektebene.....	59
3.3.4 Interkomponentenkommunikation.....	62
3.3.5 Kontext-Sharing und Shared-Spaces	63
3.3.5.1 Synchrones Sharing	63
3.3.5.2 Asynchrones Sharing durch Shared-Spaces.....	64
3.3.6 Das Konzept der Mitgliedschaft (Membership)	69
3.3.7 Kontext-Awareness	71
3.3.8 Notification.....	75
3.3.9 Single-Sign-On.....	77
3.3.10 Kontextnavigation	78
3.3.11 Metadatenmodell.....	82
 4. EVALUATION: UMSETZUNG DER KONZEPTE VON CONTEXTUAL COLLABORATIVE FRAMEWORKS AM BEISPIEL DER IBM WORKPLACE- ARCHITEKTUR	 84

4.1 IBM Workplace Architektur.....	84
4.1.1 Überblick über die IBM Workplace Produktfamilie	84
4.1.2 Überblick über die IBM Workplace Architektur	87
4.2 Umsetzung der Anforderungen.....	89
4.2.1 Komponentisierung	89
4.2.2 Komponentenintegration	91
4.2.2.1 API basierte Integration	91
4.2.2.2 Komponentenbasierte Integration.....	92
4.2.2.2 Objektbasierte Integration.....	96
4.2.3 Interkomponentenkommunikation.....	99
4.2.3.1 Interportletkommunikation	99
4.2.3.2 Interplugin-Kommunikation	102
4.2.4 Gemeinsam genutzte Plätze - Shared-Spaces	104
4.2.5 Membership.....	110
4.2.6 Kontext-Awareness	112
4.2.6.1 Online-Awareness.....	112
4.2.6.2 Place-based-Awareness	114
4.2.6.3 Device-based-Awareness.....	115
4.2.6.4 Ressource-Awareness	115
4.2.7 Notification.....	117
4.2.8 Single-Sign-on.....	119
4.2.9 Kontextnavigation	119
4.2.10 Metamodell.....	119
4.3 Zusammenfassende Bewertung.....	121
5. AUSBLICK	123
6. ZUSAMMENFASSUNG.....	125
ANHANG A.....	128
LITERATURVERZEICHNIS.....	130

Abbildungsverzeichnis

ABB. 1: KLASSIFIKATIONSSCHEMA NACH UNTERSTÜTZUNGFUNKTION	7
ABB. 2: DIE ELEMENTE DER ORGANISATION	9
ABB. 3: GESCHÄFTSPROZESS ALS KONTEXT	14
ABB. 4: ENTWICKLUNG KOLLABORATIVER SOFTWARE IM HISTORISCHEN VERLAUF	15
ABB. 5: GEMEINSAME STRUKTURELEMENTE VERSCHIEDENER PORTALIMPLEMENTIERUNGEN	21
ABB. 6: VIER-STUFENMODELL DER KONTEXTUELLEN KOLLABORATION	35
ABB. 7: DREI ARTEN VON KONTEXTBEZIEHUNGEN	42
ABB. 8: MULTIDIMENSIONALES GEFLECHT VON KONTEXTBEZIEHUNGEN	47
ABB. 9: NAVIGATIONSPFAD ÜBER KONTEXTBEZIEHUNGEN	48
ABB. 10: ARCHITEKTUR EINES IDEALTYPISCHEN FRAMEWORKS ZUR KONTEXTUELLEN KOLLABORATION	50
ABB. 11: DER GESAMTPROZESS EINER KOMPONENTENBASIERTEN APPLIKATIONSENTWICKLUNG	55
ABB. 12: INTEGRATION KOLLABORATIVER FUNKTIONEN IN DAS KONTEXTMENÜ EINER PERSON	60
ABB. 13: KONTINUUM VON AD-HOC ZU FORMALEN SPACES	66
ABB. 14: COMMUNITY UND MEMBER EINES SHARED-SPACE	70
ABB. 15: FORMEN DER KONTEXT-AWARENESS	71
ABB. 16: SCHRITTWEISE NAVIGATION ÜBER KONTEXTBEZIEHUNGEN	80
ABB. 17: SICHTENBILDUNG IN KONTEXTBEZIEHUNGEN	81
ABB. 18: WORKPLACE ARCHITEKTUR	87
ABB. 19: ANLEGEN EINES NEUEN TEAMSPACE	93
ABB. 20: IBM WORKPLACE TEMPLATE LIBRARY	93
ABB. 21: ANLEGEN EINES NEUEN TEMPLATES	94
ABB. 22: IBM WORKPLACE BUILDER – EIGENSCHAFTEN UND POINTS OF VARIABILITY	95
ABB. 23: KOLLABORATIVE FUNKTION INTEGRIERT IN DAS KONTEXTMENÜ EINER PERSON	97
ABB. 24: INTERPORTLETKOMMUNIKATION MIT HILFE DES PROPERTY BROKER	101
ABB. 25: IBM WORKPLACE ACTIVITY EXPLORER	105
ABB. 26: MEMBERSHIP-PORTLET MIT ROLLEN IM TEAMSPACE	110
ABB. 27: INSTANT CONTACTS MIT ONLINE-AWARENESS-FUNKTIONALITÄT	113
ABB. 28: PLACE-BASED-AWARENESS IN EINER WEB-KONFERENZ	115
ABB. 29: RESSOURCE-AWARENESS IN EINER ACTIVITY	116
ABB. 30: EIN ALERT AN DIE MITGLIEDER EINER ACTIVITY	118

Tabellenverzeichnis

TABELLE 1: COLLABORATION CATEGORIES [CAIN 2003 S.4]	18
TABELLE 2: OBJEKTTYPEN UND BEZIEHUNGSTYPEN DER KONTEXTUELLEN KOLLABORATION.....	78
TABELLE 3: ZUSAMMENFASSENDE BEWERTUNG DER IBM WORKPLACE ARCHITEKTUR.....	121

Abkürzungsverzeichnis

AOL	America Online
API	Application Programming Interface
CRM	Customer Relationship Management
CSCW	Computer Supported Cooperative Work
EJB	Enterprise Java Bean
ERP	Enterprise Resource Planning
IBM	International Business Machines
IDC	International Data Corporation
IM	Instant Messaging
IP	Internet Protocol
JSR	Java Specification Request
PDA	Personal Digital Assistant
PoV	Point of Variability
RSS	Really Simple Syndication
SCM	Supply Chain Management
SDK	Software Development Kit
SIP	Session Initiation Protocol
SMS	Short Message Service
SOA	Service Oriented Architecture
SPI	Service Provider Interface
UI	User Interface
URL	Universal Resource Locator
VoIP	Voice over IP
WMC	Workplace Managed Client
WCS	Workplace Collaboration Services

1. Einleitung

1.1 Motivation der Arbeit

Elektronische Kommunikations- und Kollaborationswerkzeuge wie E-Mail, Instant Messaging und Web-Konferenzen haben insbesondere seit Mitte der 90er Jahre starke Verbreitung in Unternehmen gefunden und die Kommunikations- und Kollaborationsmöglichkeiten der Mitarbeiter grundlegend verändert. Insbesondere für verteilt und global operierende Unternehmen haben diese unter dem Begriff „Groupware“ zusammengefassten Applikationen eine verbesserte Zusammenarbeit in global verteilten und virtuellen Teams gebracht oder teilweise sogar erst ermöglicht.

Die zunehmende Globalisierung bringt den Unternehmen aber nicht nur neue Möglichkeiten sondern vor allem auch einen stetig wachsenden Wettbewerbsdruck, dem sie sich stellen müssen, um am Markt zu bestehen. Eine stetig wachsende Markttransparenz und die zunehmende Vergleichbarkeit von Produkten und Leistungen haben auch zu immer höheren Anforderungen der Kunden geführt. Die Kunden erwarten eine immer größere Flexibilität der Hersteller. Preis und Qualität sind heute als alleinige Wettbewerbsfaktoren nicht mehr ausreichend um sich am Markt behaupten zu können. Die Unternehmen müssen vor allem auch bezüglich Service, Liefergeschwindigkeit und der Erfüllung spezifischer Kundenwünsche überdurchschnittliche Leistungen erbringen.

Dieser Trend hat bei den Unternehmen seit einiger Zeit dazu geführt, dass sie sich verstärkt auf die Optimierung und Flexibilisierung ihrer Geschäftsprozesse besinnen. Durch die Notwendigkeit, auf ständig wechselnde Kundenwünsche reagieren zu können, steigt der Bedarf zur Kommunikation und Koordination der Mitarbeiter im Rahmen der Ausführung von Geschäftsprozessen. Eine effiziente Integration von kollaborativen Funktionen in die Bearbeitung der Geschäftsprozesse wird daher als ein wesentlicher Faktor genannt, um das Prozessergebnis nachhaltig zu verbessern [vgl. Gotta 2004, S.3].

Bisher ist die Kollaborationsstrategie der Unternehmen allerdings meist Tool-orientiert und von den Business-Applikationen getrennt. Kollaborationswerkzeuge wie E-Mail, Instant Messaging oder Web-Konferenzen stehen als separate Applikationen oder als integrierte Groupwaresysteme neben den produktiven Business-Applikationen wie Customer-Relationship-Management (CRM) oder Enterprise-Resource-Plannig (ERP) Systemen. Will ein Mitarbeiter im Rahmen eines Geschäftsprozesses mit einem Kollegen kollaborieren, so muss er den Arbeitskontext seiner vertrauten Business-Applika-

tion verlassen, um ein Kollaborationswerkzeug zu benutzen. Dadurch entstehen Medienbrüche und Ineffizienzen, die nicht zur Verbesserung des Prozessergebnisses beitragen.

Unternehmen fangen daher gerade an, ihre Kollaborationsstrategie zu überdenken. Anstatt elektronische Kollaboration als Mittel zur Verbesserung persönlicher Produktivität oder einzelner Aufgaben anzusehen, wird der Wert der Kollaboration daran gemessen, in welcher Weise sie zur Bearbeitung und der Verbesserung des Geschäftsprozessergebnisses beiträgt [vgl. Gotta 2004, S.2]. Dazu sollen die derzeitigen, monolithischen Kollaborationssysteme in einzelne, funktionale Einheiten aufgeteilt werden, die dann durchgängig in die operativen Business-Applikationen integriert werden um so die Kollaboration aus dem Kontext des Geschäftsprozesses zu ermöglichen.

Dieser Ansatz wird aktuell unter dem Begriff „**kontextuelle Kollaboration**“ (Contextual Collaboration) diskutiert.

1.2 Aufbau und Zielsetzung der Arbeit

Die vorliegende Arbeit gibt einen Überblick über die Konzepte und Anforderungen der kontextuellen Kollaboration. Ziel ist es, diese Anforderungen zu systematisieren und daraus funktionale und strukturelle Systemmerkmale abzuleiten und zu beschreiben. Diese Systemmerkmale sollen in Form eines idealtypischen Frameworks zusammengefasst werden, welches die Umsetzung kontextuell-kollaborativer Systeme ermöglicht und damit zur effizienteren Kollaboration im Kontext von Geschäftsprozessen führt.

Im zweiten Kapitel werden dazu die begrifflichen Grundlagen gelegt und unter anderem auf die Konzepte der kontextuellen Kollaboration und der Geschäftsprozesse eingegangen. Das dritte Kapitel stellt dann ein fiktives aber praxisnahes Szenario vor, welches die verschiedenen Aspekte der kontextuellen Kollaboration im praktischen Einsatz verdeutlicht. Daraufhin wird das im Rahmen dieser Arbeit entwickelte Vier-Stufenmodell der kontextuellen Kollaboration vorgestellt, welches die Anforderungen an ein Framework für kontextuelle Kollaboration systematisiert und daraus Systemmerkmale herleitet. Diese Systemmerkmale werden im Anschluss in Form eines idealtypischen Frameworks detailliert beschrieben und diskutiert. Im Rahmen dieser Übersichtsarbeit wird dazu zunächst von speziellen Plattformtechnologien und Herstellern abstrahiert. Das vierte Kapitel evaluiert und bewertet dann am konkreten Beispiel der *IBM Workplace*

1. Einleitung

Architektur, wie und in welchem Umfang diese Systemmerkmale in der Praxis umgesetzt werden. Mit dem fünften Kapitel schließt die Arbeit in einer zusammenfassenden Bewertung ab.

2. Grundlagen

Im Mittelpunkt des Konzeptes von „Contextual Collaboration“ (dt: „kontextuelle Kollaboration“) steht die elektronisch unterstützte Kollaboration. Wie schon in der Einleitung kurz skizziert, bezeichnet der Begriff „Contextual Collaboration“ zunächst ein Konzept, welches Mitarbeitern und Kunden ermöglichen soll, sämtliche elektronisch unterstützten Kollaborationsfunktionalitäten wie z.B. E-Mail, Chat, Video-Konferenzen, Workflows etc. zu nutzen und zwar innerhalb ihrer operativen Business-Applikationen. Das bedeutet, computergestützte Kollaboration findet im Kontext der (täglichen) Geschäftsvorfälle statt. Dabei zielt eine grundsätzliche Konzeptstrategie darauf, Kollaborationsfunktionalitäten als Komponenten zu betrachten und sie in jede beliebige Business-Applikation integrierbar zu machen.

Elektronisch unterstützte Kollaboration hat eine zentrale Bedeutung in aktuellen Geschäftsprozessen. Es handelt sich bei Contextual Collaboration nicht um eine grundsätzlich neue Softwaretechnologie, sondern um einen neuen Ansatz, diese Geschäftsprozesse besser elektronisch zu unterstützen. Dabei werden bestehende Technologien und Software-Funktionalitäten genutzt. „Rather than view collaboration as a personal productivity endeavor (e.g., “saving time,” making individual tasks more efficient), the goal here is to enable the process to perform at a higher level. [...] Collaboration is viewed as successful based on the process outcome, not the output of individuals within various tasks.” [Gotta 2004, S.2]

In diesem Kapitel werden die Grundlagen von computergestützter Kollaboration und des Begriffs „Geschäftsprozess“ erläutert. Es wird eine Begriffsdefinition für „Contextual Collaboration“ hergeleitet und erklärt. Dabei werden unterschiedliche Aspekte dieses Konzeptes dargestellt.

Da die im vierten Kapitel vorgestellte und evaluierte *IBM Workplace* Architektur auf einer Portaltechnologie basiert, wird in Kapitel 2.4 eine Einführung in die Grundlagen der Portaltechnologie gegeben.

2.1 Contextual Collaboration als Kollaborationskonzept in CSCW und Groupware

Der englische Begriff „Contextual Collaboration“ ist kein von der Wissenschaft geprägter Begriff, sondern entstammt dem Umfeld der Industrie und Wirtschaft. „The concept of contextual collaboration has been spearheaded by industry and covered in the trade press, but has been largely overlooked by research community.“ [Hupfer et al. 2004, S.1] Matthew Cain, ein Analyst der Gartner Group prägte diesen Namen. Auch IDC bezeichnen sich als die Erfinder des Begriffes. [vgl. Mahowald 2001, S.2]

Auch wenn der Begriff nicht wissenschaftlich begründet wurde, so lässt er sich inhaltlich aber dem wissenschaftlichen Forschungsgebiet der „Computer Supported Cooperative Work“ (CSCW) zuordnen. „Erklärtes Ziel [der CSCW] ist es, die Zusammenarbeit von Menschen durch den Einsatz von Informations- und Kommunikationstechniken zu verbessern, d.h. sowohl effizienter und flexibler als auch humaner und sozialer zu gestalten.“ [Nastansky et al. 2000, S.240]

Grundsätzlich handelt es sich bei Contextual Collaboration um ein spezielles Konzept der computergestützten Kollaboration. Daher wird zunächst ein Überblick über das wissenschaftliche Verständnis des Teilbegriffs „**Collaboration**“ im Rahmen dieser Arbeit gegeben.

Im Deutschen wird in der Regel nicht die direkte Übersetzung „Kollaboration“ für die gleiche Semantik verwandt. Kollaboration wird eher als „Zusammenarbeit mit dem Feind“ verstanden und ist im deutschen Sprachraum negativ konnotiert. Daher wird häufig der deutsche Begriff „Kooperation“ mit dem englischsprachigen Begriff Collaboration gleich gesetzt. [Gesellschaft f. Informatik 2006] Um allerdings die Nähe zu dem Kernbegriff „Contextual Collaboration“ auch im Deutschen sprachlich beizubehalten, wird hier „Contextual Collaboration“ mit „kontextuelle Kollaboration“ übersetzt. Beide Begriffe werden im Rahmen dieser Arbeit synonym verwendet.

„Die Kooperation baut auf der Dimension der Kommunikation auf und stellt den Austausch von Informationen mit einem gemeinsamen Ziel dar. Kooperation bedingt, dass mindestens zwei Personen in einen gemeinsamen zielgerichteten Kooperationsprozess involviert sind.“ [Dierker/Sander 1998, S.104f] Die Dimension „Kommunikation“ ist folglich unabdingbar für die Kollaboration. Kommunikation als Voraussetzung hat die Schlüsselrolle in der teamorientierten Arbeit. Als Kommunikation wird die einseitige

2. Grundlagen

oder wechselseitige Übertragung von Informationen zwischen Mitgliedern einer Gruppe bezeichnet. Kommunikation wird als Grundlage für Kollaboration und Koordination gesehen.

Eine weitere hier genannte Unterstützungsfunktion, die Koordination, ist folglich eng verknüpft mit Kollaboration und Kommunikation. „Die Koordination verfolgt das Ziel, die Kooperation von Gruppenmitgliedern so zu steuern, dass das angestrebte Ziel möglichst effizient erreicht werden kann.“ [Bruse 2005, S.47] Workflow-Systeme z.B. dienen der Unterstützung der Koordination.

Im Rahmen dieser Arbeit wird der Begriff Kollaboration als ein Oberbegriff der drei oben erläuterten Unterstützungsfunktionalitäten Kommunikation, Kooperation und Koordination verstanden und vereint ihre Merkmale in sich.

"Collaboration is broadly defined as the interaction among two or more individuals and can encompass a variety of behaviors, including communication, information sharing, coordination, cooperation, problem solving, and negotiation." [Hall, T. 1999, S.10]

Groupware

„Während CSCW der wissenschaftstheoretische Ansatz ist, bezeichnet Groupware eine Klasse von Software-Produkten, die die Arbeit von Gruppen unterstützen.“ [Bruse 2005, S.45] Abb.1 zeigt die vier Groupware-Systemklassen von Unterstützungsfunktionen. Diese hier gezeigten Systemklassen bilden zusammen die Grundlage für die Einordnung der Kollaborationsunterstützungswerkzeuge, die bei der kontextuellen Kollaboration zum Einsatz kommen können.

Andere, hier nicht gezeigte Unterstützungswerkzeuge lassen sich den Systemklassen zuordnen. Im Rahmen dieser Arbeit werden die Kollaborationsunterstützungswerkzeuge der vier Klassen als „kollaborative Funktionalitäten“ bezeichnet. Dabei werden neben den klassischen kollaborativen Funktionalitäten wie Email oder Instant Messaging auch neuere Kollaborationswerkzeuge, wie Blogs, Wikis oder Voice over IP als für die kontextuelle Kollaboration geeignete Werkzeuge angesehen. Es werden daher keine Unterstützungswerkzeuge für die kontextuelle Kollaboration per se ausgeschlossen.

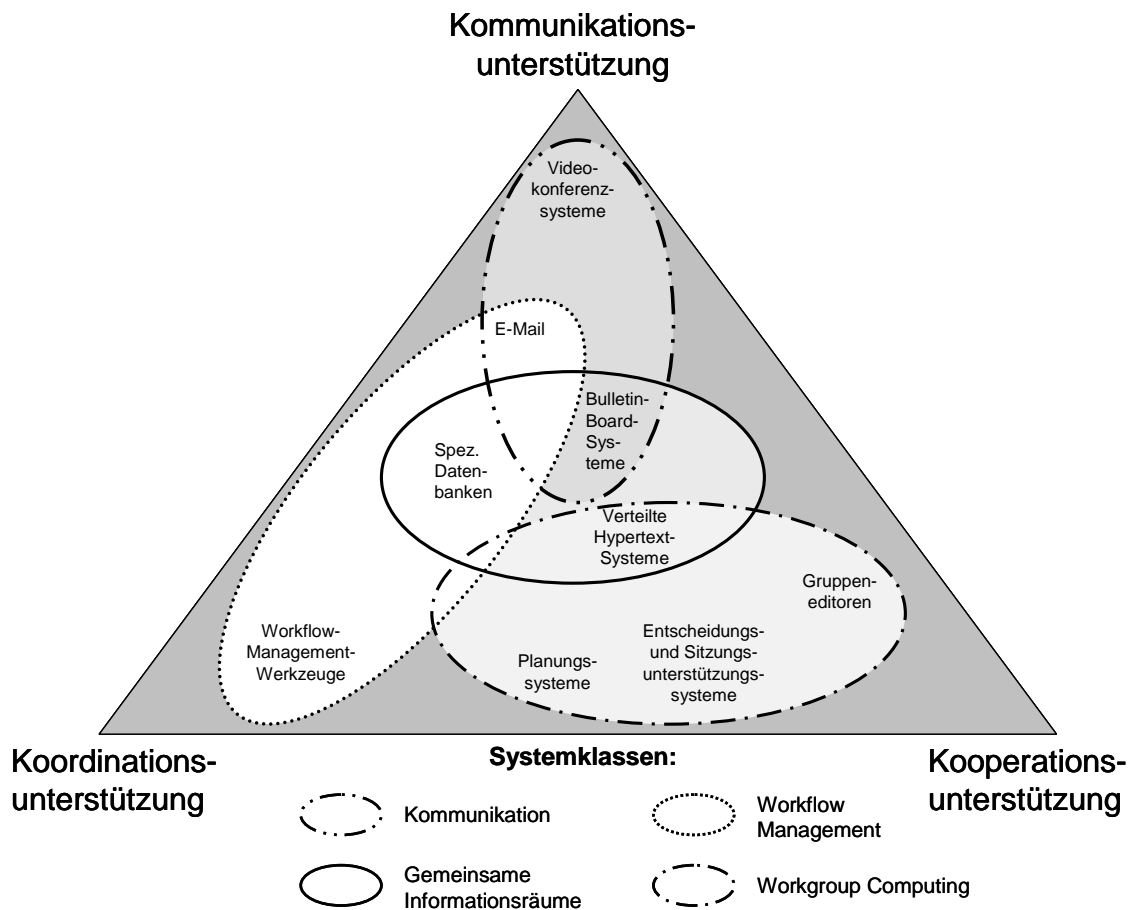


Abb. 1: Klassifikationsschema nach Unterstützungsfunktion

[Nastansky et al. 2000, S.243]

2.2 Geschäftsprozesse

Ein Merkmal kontextueller Kollaboration ist, dass der Geschäftsprozess (engl.: Business-Process) als der Ausgangspunkt der kollaborativen Aktivität betrachtet wird. Kontextuelle Kollaboration wird als ein Faktor angesehen, der das Prozessergebnis verbessern kann. „Collaboration strategies are shifting from a focus on tools to an eye on improving process performance. Improving work practices within business processes requires collaboration ‘in context’ while tapping into communities of practice as sources of best practices and innovation.” [Gotta 2004, S.1] Wie gut die Business-Prozesse ablaufen, ist entscheidend für den Erfolg von Unternehmen. Sie erwirtschaften die so Wertschöpfung. [vgl. Huang 2004, S. 69]

Ein Geschäftsprozess ist definiert als: „a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.” [WfMC 1999, S.10]

2. Grundlagen

Geschäftsprozesse sind demnach an eine gewisse Zielsetzung und an Geschäftsgebiete gebunden: sie können in ihrer Größe variieren, d.h. sie können innerhalb einer Organisationseinheit stattfinden oder auch ein unternehmensweites Ausmaß haben. Geschäftsprozesse haben darüber hinaus häufig einen gewissen Wiederholungscharakter. [vgl. Huang 2004, S.69]

Im Rahmen dieser Arbeit wird aber auch ein einmaliger zielgerichteter Vorgang als Geschäftsprozess angesehen. Die Wiederholung gilt hier nicht als zwingend. Nicht nur strukturierte Prozesse, die einer strengen Abfolge gehorchen werden hier betrachtet, sondern auch die ad hoc initiierten und unstrukturierten Vorgänge gehören der Gesamtheit der hier betrachteten Prozesse an. Es wird daher unter Geschäftsprozess nicht nur der im Sinne von [Nastansky et al. 2000, S.288] verwendete Begriff Workflow gemeint, sondern der Begriff schließt hier auch die unstrukturierten und schlecht automatisierbaren Prozesse mit ein. „Business processes are increasingly being automated by information technology, but these are formalized processes that have limited adaptability. While the semantic eBusiness vision can enhance business processes and business information exchange, it is people who conduct the non-routine, value-added work that cannot be automated. Most knowledge work is collaborative, informal and situationally-adaptive.“ [Moran 2005, S.1]

[Huang 2004, S.69.] definiert weiter: „Ein Geschäftsprozess beinhaltet die Gesamtheit und die Aufeinanderfolge von Arbeitsschritten, die zusammengenommen eine Leistung erbringen.“ Im Sinne von kontextueller Kollaboration ist der Geschäftsprozess aber mehr als nur die Summe der einzelnen individuellen Leistungen, sondern das Prozessergebnis wird erheblich von der Kollaborationsleistung der Mitarbeiter bestimmt. „This design model is different from earlier collaboration efforts that focused on people and not how people work within processes.“ [Gotta 2004, S.3] Wie gut diese Kollaborationsleistung ist, hängt von den Kollaborationsstrategien der Unternehmen ab. Kontextuelle Kollaboration wird als eine Kollaborationsstrategie angesehen, die diese Kollaborationsleistung und damit auch das Prozessergebnis verbessern kann. „Our research indicates overwhelming preference by senior management that collaboration strategies evolve from a focus on personal productivity to address process performance and innovation.“ [Gotta 2004, S.1]

Im Umfeld der kontextuellen Kollaboration ist es außerdem wichtig, dass zu der oben genannten „Gesamtheit von Arbeitsschritten“ nicht nur die direkten Prozessdokumente

2. Grundlagen

und Prozessergebnisse als solches gehören, also z.B. das fertige Bestellformular bei einem Bestellprozess, sondern dass zu dieser Gesamtheit auch alle kollaborativen Vorgänge wie Mails, Instant-Messaging-Kommunikation oder die Video-Konferenz und alle mit dem Prozess in Beziehung stehenden Informationen gehören. Zu diesen Informationen zählt auch das indirekte Prozesswissen, z.B. welche Personen sind am Prozess beteiligt, wer ist der Hauptverantwortliche usw. Als eine Aufgabe von kontextueller Kollaboration wird im Rahmen dieser Arbeit verstanden, dass diese Kollaborationsvorgänge nicht nur im Kontext der Geschäftsprozesse initiiert werden, sondern dass sie und auch sämtliche Informationen im Kontext der Geschäftsprozesse referenziert werden und dadurch persistent gemacht werden.

Bei der Bearbeitung von Geschäftsprozessen entstehen folglich Beziehungen zwischen den beteiligten Elementen, die gemeinsam in einem Aufgabenerfüllungskontext stehen. [vgl. Huang 2004, S.106]

Schmitz identifiziert vier verschiedene Elemente einer Organisation: Aufgabe, Aufgabenträger, Sachmittel, und Information. Das Zusammenspiel dieser Elemente beschreibt im Rahmen des Aufgabenerfüllungskontextes welche Personen (wer), auf welche Art (wie) welchen Prozess (was) mit welchen Mitteln bearbeiten. Im Kontext dieser Diplomarbeit werden diese Elemente als Dimensionen der Aufgabenerfüllung bezeichnet.

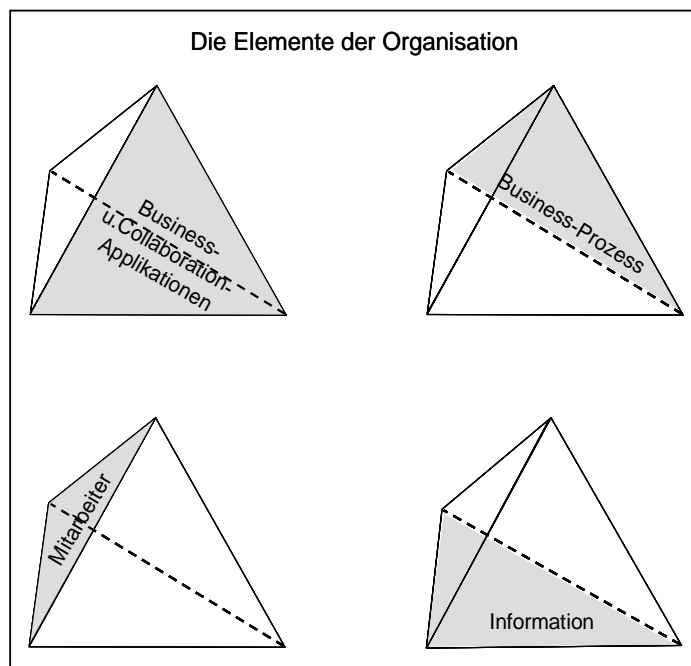


Abb. 2: Die Elemente der Organisation
[in Anlehnung an Schmitz 1991]

„Das Kernelement der Organisation ist die Aufgabe. Wenn es keine Aufgabe gibt, sind auch alle übrigen Elemente (Aufgabenträger, Sachmittel, und Informationen) entbehrlich. Insofern nimmt die Aufgabe eine ganz besondere Stellung ein.“ [Schmitz 1991, S.18] Als Synonym für „Aufgabe“ im Sinne von Schmitz wird der in dieser Arbeit verwendete Begriff „**Geschäftsprozess**“ oder „Business-Prozess“ angesehen.

„Aufgabenträger sind Menschen, die eingesetzt werden, um Aufgaben zu bewältigen.“ [Schmitz 1991, S.18] Aufgabenträger werden in dieser Arbeit in der Regel als **Mitarbeiter**, Personen oder auch Anwender bezeichnet.

„Sachmittel unterstützen die Aufgabenerfüllung.“ [Schmitz 1991, S.18] In diesem Sinne werden hier die unterschiedlichen Tools, also sowohl die operativen **Business-Applikationen** als auch die **kollaborativen Funktionalitäten** als die für diese Arbeit relevanten Sachmittel von Unternehmen angesehen.

„Die Bewältigung von Aufgaben setzt **Informationen** voraus.“ [Schmitz 1991, S.18] Informationen sind alle Daten, die den Geschäftsprozess betreffen. Im Sinne dieser Arbeit ist dies z.B. ein Angebot oder eine Rechnung für einen Kunden. Es schließt aber auch das implizite Wissen der Mitarbeiter über einen Business-Prozess oder den verbalen Informationsaustausch über den Prozess mit ein.

2.3 Kontextuelle Kollaboration als Begriff

Wie in Kapitel 2.1 erläutert gilt Matt Cain als Begründer des Begriffs „Contextual Collaboration“ (Kontextuelle Kollaboration). Er versteht Contextual Collaboration folgendermaßen: „Collaboration capabilities will achieve their true potential when they are embedded directly into applications, as opposed to residing in a separate program or infrastructure, a concept we call contextual collaboration. To achieve contextual collaboration, technologies are required that enable users to instantly collaborate from within programs they are already familiar with and use every day.“ [Advanced Reality 2002, S.1]

Der Duden bezeichnet Kontext mit *Zusammenhang* oder *Verknüpfung* [vgl. Duden 2004] Das Konzept „Contextual Collaboration“ wird als „contextual“ bezeichnet, weil die Kollaboration hier nicht als unabhängiger, in sich verkapselter Vorgang betrachtet wird, sondern weil die Kollaboration in einem Kontext stattfindet. „Contextual“ ist dann im Sinne von „mit Kontext verknüpfen“ zu verstehen. Damit bedeutet „Contextual Col-

laboration“ vom Begriff her zunächst ganz allgemein: „Kollaborieren in einem Kontext“ bzw. „Kollaborieren in einem direkten Zusammenhang“.

Im Folgenden wird das kollaborative Konzept definiert und dargelegt. Eine genauere Schematisierung und Erläuterung findet in Kapitel drei statt.

2.3.1 Der Geschäftsprozess als Kontext der kontextuellen Kollaboration

Als Synonyme für Kontext werden häufig „Umgebung“ oder „Situation“ genannt. Dey und Abowed definieren Kontext auch im Sinne von Information über Situationen. „Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.“ [Dey/Abowed 1999, S.3] „Contextual Collaboration“ kann somit auch als situationsbezogene Kollaboration bezeichnet werden. So interpretieren auch Chen et al. das Verständnis von Cain von Contextual Collaboration. „The Type of communication Cain envisioned was circumstance-dependent, where an event occurred depending on the setting“ [Chen et al. 2004, S.4].

In der Aktivitätentheorie wird eine Aktivität selbst als Kontext betrachtet. „Activity Theory, then, proposes a very notion of context: the activity itself is the context. What takes place in an activity system composed of object, actions and operation is the context. Context is constituted through the enactment of an activity involving people and artifacts. Context is not an outer container or shell inside of which people behave in certain ways. People consciously and deliberately generate contexts (activities) in part through their own objects; hence context is not just ‘out there’.“ [Nardi 1996, S.76]

In Anlehnung an die Definition von Kontext durch die Aktivitätentheorie, so wie Nardi sie beschreibt, ist der Kontext also nicht etwas was die tägliche Arbeit umgibt, sondern die tägliche Arbeit selbst ist der Kontext der Kollaboration. Damit bedeutet Contextual Collaboration: „Kollaborieren im Kontext der täglichen Arbeit“.

Cain und auch andere Analysten wie z.B. Mahowald haben bei der Prägung und Verwendung des Begriffes allerdings eine sehr konkrete Vorstellung von dem Konzept „Contextual Collaboration“ in der praktischen Umsetzung. Für sie bedeutet „Contextual Collaboration“ ein Kollaborationskonzept in dem die Kollaborationsfunktionalitäten keine abgeschlossenen Applikationen sind, sondern integrierte Funktionalitäten in Ap-

plikationen der täglichen Arbeit. [vgl. Mahowald 2001, S.1] Wie bereits in Kapitel 2.2 erläutert wurde, manifestiert sich die tägliche Arbeit der Zielgruppe kontextueller Kollaboration, dies sind vor allem Knowledge Worker, überwiegend in Geschäftsprozessen. Damit wird der Geschäftsprozess zu dem Kontext selbst, in dem kollaboriert wird. D.h. der Begriff „Kontextuelle Kollaboration“ bezeichnet das Konzept der Kollaboration im Kontext von Geschäftsprozessen. Da die Geschäftsprozesse in der Regel in operativen Business-Applikationen bearbeitet werden, sich also dort manifestieren, hat die Definition aus Sicht von Unterstützungssystemen die Bedeutung von „Kollaborieren im Kontext der Business-Applikationen“. In diesem Sinne wird für Contextual Collaboration auch der Begriff „In-Context Collaboration“ verwendet. „In-Context Collaboration: [...] In this scenario, I am operating in an environment where the real-time collaborative capabilities of instant messaging and web conferencing are not isolated in standalone applications, but instead are brought into the context of my work.“ [Boyd 2003, S.2]

Dabei sollen die Kollaborationsfunktionalitäten als Service-Komponenten zur Verfügung stehen um so in den Kontext der Geschäftsprozesse integriert werden zu können. Integration von Kollaborationskomponenten ist somit als einem Hauptmerkmal der kontextuellen Kollaboration anzusehen. [vgl. Huang 2004, S.106]

Es ergibt sich als Arbeitsgrundlage für diese Arbeit zusammenfassend folgende Definition von kontextueller Kollaboration:

„Kontextuelle Kollaboration bezeichnet ein Kollaborationskonzept für Unternehmen, welches es Mitarbeitern, Kunden und Personen anderer Unternehmen ermöglicht, geschäftsprozessorientiert zusammenzuarbeiten. Dazu werden die zur Kollaboration genutzten Unterstützungsfunktionen als integrierbare Komponenten zur Verfügung gestellt und halten die Fähigkeit bereit, im Kontext eines bestimmten Geschäftsprozesses zu kollaborieren und sich auf einen Geschäftsprozess zu kontextualisieren. Der Geschäftsprozess vereinigt dabei alle zum Aufgabenerfüllungskontext gehörenden Beziehungen auf sich.“

Bei dem Konzept „Contextual Collaboration“ geht es folglich nicht um die Automatisierung von Kollaboration und Kommunikation. Die Kommunikation soll nicht von der Technologie übernommen werden, d.h. die Menschen sollen nicht aus dem Kommuni-

kations- und Kollaborationsprozess ausgeschlossen werden, sondern die Kommunikation und Kollaboration unter Mitarbeitern soll im Sinne einer besseren Teamarbeit und eines verbesserten Prozessergebnisses optimiert werden. „The goal of contextual collaboration is to make online collaboration as simple and intuitive as it is to work with people in the same room, while enabling that capacity between people anywhere in the world.“ [Chen et al. 2004, S.4] Teams soll die Möglichkeit gegeben werden die Kollaborationstechnologien zu nutzen und dadurch das Teamergebnis zu verbessern. [vgl. Chen et al. 2004, S.6] Kontextuelle Kollaboration adressiert auch nicht die vollautomatisierbaren kollaborationsarmen Geschäftsprozesse sondern es stehen die kollaborationsintensiven Prozesse im Mittelpunkt (vgl. hierzu auch Kapitel 2.2).

2.3.2 Kontextinformationen und Kontextwissen

Wie oben erläutert, ist der Geschäftsprozess, an dem ein Anwender gerade arbeitet, folglich sein aktueller Arbeitskontext.

Um diese tägliche Arbeit erledigen zu können, benötigt der Anwender Informationen. Diese zu seinem Arbeitskontext gehören Informationen werden im Rahmen dieser Arbeit als **Kontextinformationen** bezeichnet. „Contextual information is always information that is identified in relation to something else that is in the primary focus of our attention.“ [Schiffrin 1994, S. 363] Kontextinformationen können z.B. getroffene Absprachen zwischen Mitarbeitern zu einem speziellen Problem innerhalb des Geschäftsprozesses sein, oder ein Dokument, welches für die Bearbeitung des Geschäftsprozesses wichtig ist. Das bedeutet: Kontextinformationen können sowohl schon bestehende Informationen als auch Artefakte sein, die während des Kollaborationsprozesses entstehen. Diese Kontextinformationen werden bisher meistens nicht im Kontext des Geschäftsprozesses abgespeichert sondern verbleiben in den entsprechenden Applikationen, d. h. etwa eine E-Mail in der Mailbox des Mitarbeiters.

Dabei wird im Rahmen dieser Arbeit auch spezielles Kontextwissen über die Geschäftsprozesse als Kontextinformationen betrachtet. Kontextwissen ist oftmals kein explizites Wissen, sondern manifestiert sich als implizites Wissen durch Erfahrung und gespeicherte Information in den Köpfen der Mitarbeiter. [Huth 2004, S.37] Es hat sich gezeigt, dass das Kontextwissen eines Business-Prozesses von entscheidender Bedeutung für ein positives Prozessergebnis sein kann. Ein Ziel von kontextueller Kollabora-

tion ist es, die Kontextinformationen und das Kontextwissen auch im Kontext des Geschäftsprozesses zu erhalten.

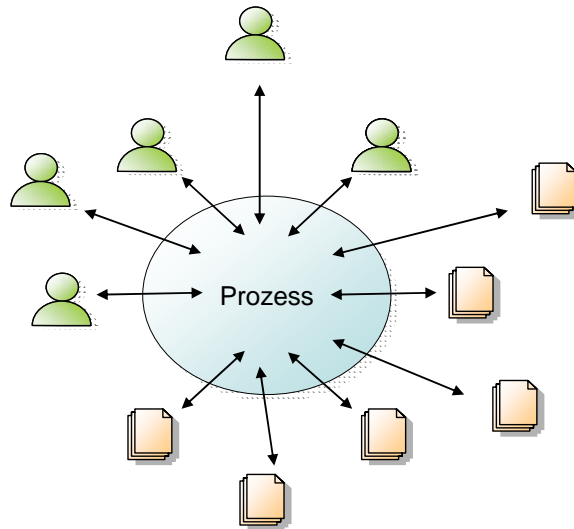


Abb. 3: Geschäftsprozess als Kontext

Ein weiteres Ziel ist es, implizites Wissen über Prozesse in explizites Wissen umzuwandeln. Durch diese Umwandlung wird erreicht, dass das Wissen instrumentalisierbar gemacht wird und im Kontext der Geschäftsprozesse abgespeichert werden kann. Z.B. kann eine Diskussion über Softwaredesignentscheidungen, die in einem Team als E-Meeting geführt wurde, im Kontext der Softwarespezifikation abgespeichert werden. Auf diese Weise ist es möglich, Ergebnisse für nachfolgende Projektmanager und andere Teammitglieder nachvollziehbar festzuhalten. Dieses Vorgehen der Kontextualisierung, also der Beziehungsherstellung zwischen Dokumenten und Kontext kann dem Anwender ein fokussierteres Arbeiten ermöglichen.

2.3.3 Historische Entwicklung kollaborativer Systeme

Der Ansatz der kontextuellen Kollaboration hat sich in den letzten Jahren aus einer Reihe von Entwicklungsschritten speziell auf dem Gebiet der kollaborativen Softwaresysteme und der Entwicklung von Softwarearchitekturen im Allgemeinen herausgebildet.

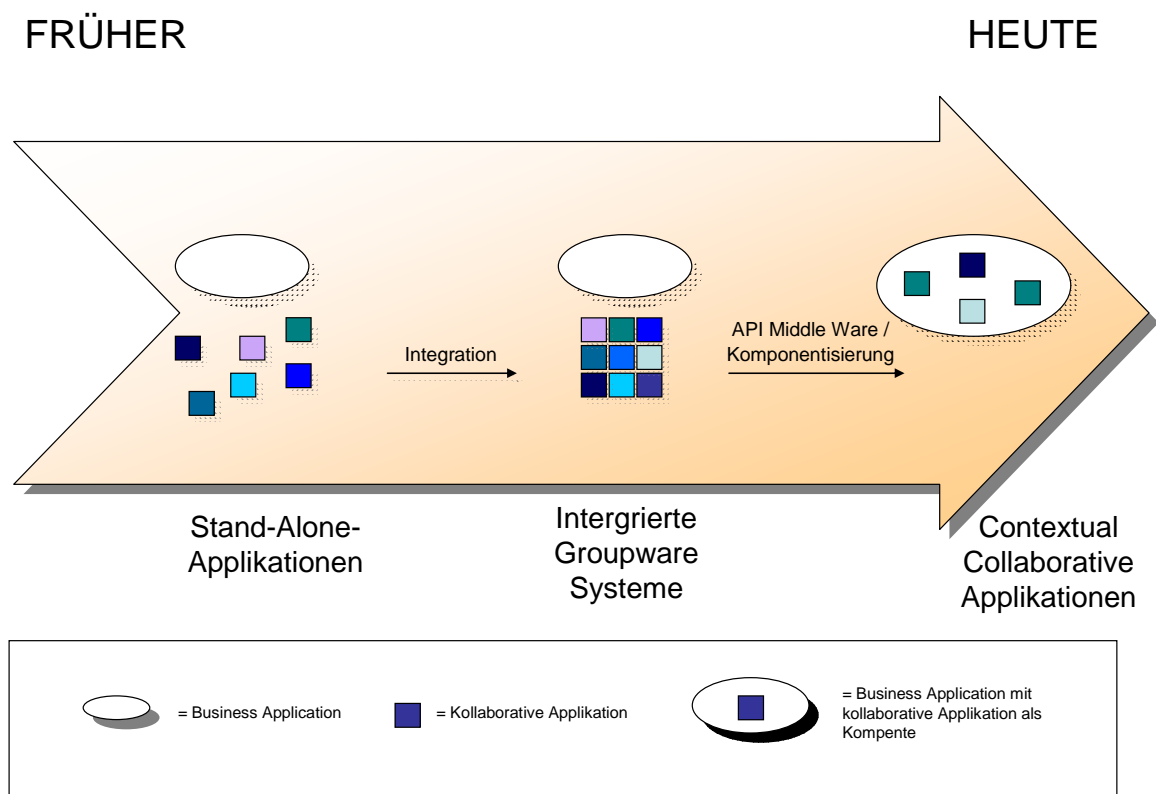


Abb. 4: Entwicklung Kollaborativer Software im historischen Verlauf

Zunächst wurden kollaborative Werkzeuge in Form von separaten **Stand-alone-Applikationen** entwickelt. E-Mailprogramme zählen dabei zu den ältesten und am weitesten verbreiteten Systemen. Newsgroup Reader, Instant Messaging, Chat Clients, Video-Konferenzen oder Dokumentenmanagementsysteme sind weitere Beispiele.

Mit steigender Anzahl verschiedener, untereinander inkompatibler Kommunikations- und Kollaborationswerkzeuge unterschiedlicher Hersteller und der steigenden Verbreitung dieser Applikationen im Umfeld von Unternehmen, wuchs im Laufe der 90er Jahre der Bedarf an stärker integrierten Kollaborationssystemen, die eine Reihe unterschiedlicher Kollaborationsfunktionen in sich vereinigten. Für diese integrierten Systeme hat sich der Begriff „**Groupwaresysteme**“ geprägt [vgl. Kapitel 2.1]. Lotus Notes/Domino oder GroupWise von Novell sind Beispiele für diese Softwaregattung. Diese integrierten Systeme zeichnen sich durch gemeinschaftlich genutzte Services aus, wie z.B. ein gemeinsames User-Directory, gemeinsam genutzte Datastores, und gemeinsame Replikationsmechanismen. Damit konzentrieren sich viele Kollaborationsfunktionen in einem integrierten, oft monolithischen Softwarepaket. Diese Systeme standen allerdings immer noch neben den operativen Business-Applikationen wie z.B. SAP oder PeopleSoft.

Der Bedarf, die gewohnten kollaborativen Funktionen als Teil operativer Business-Applikationen zu nutzen führte dazu, dass einige Groupwaresysteme erlaubten, auf ihrer Basis Applikationsentwicklungen durchzuführen. Dadurch ergab sich eine neue Art von formular- und dokumentenbasierten Applikationen die sehr kollaborationsfokussiert waren. In diesem Ansatz zeigte sich bereits, dass man die kollaborativen Funktionalitäten nicht als separate Applikationen nutzen wollte sondern als integrierte Funktionalitäten in den Business-Applikationen zur Verfügung haben wollte.

Im Zuge einer Öffnung dieser integrierten kollaborativen Systeme auch für Dritthersteller begannen einige Hersteller ihre kollaborativen Funktionalitäten in Form von Application Programming Interfaces (APIs) zur Verfügung zu stellen. So entwickelte sich eine umfangreiche Middleware Infrastruktur für synchrone und asynchrone Kollaboration, die von Drittherstellern benutzt wurde, um ihre kundenspezifischen oder serienmäßigen Business-Applikationen mit kollaborativen Funktionen anzureichern. Diese API basierte Integration durch die Hersteller von Business-Applikationen war ein erster Schritt zur kontextuellen Integration von kollaborativen Funktionalitäten in Business-Applikationen.

Heute werden im Rahmen der aktuell diskutierten **kontextuellen Kollaboration** kollaborative Funktionalitäten zunehmend als höher aggregierten und wieder verwendbare Services und Komponenten zur Verfügung gestellt. Diese erlauben es, durch die Integration kollaborativer Funktionalitäten in bestehende Business-Applikationen oder durch die einfache Zusammenstellung verschiedener Komponenten zu neuen Business-Applikationen, eine durchgängige Strategie der kontextuellen Kollaboration im Unternehmen zu etablieren.

2.3.4 Rahmenbedingungen für kontextuelle Kollaboration

Um kontextuelle Kollaboration in einem Unternehmen etablieren zu können, bedarf es nicht nur einer kontextuell-kollaborativen Architektur aus Soft- und Hardware sondern auch organisatorischer und unternehmenskultureller Maßnahmen.

Nach Cains Meinung sollten Unternehmen zunächst feststellen, in welchem Kollaborationsstadium, der „Collaboration Maturity“, [vgl. Cain 2003, S.2] sie sich zurzeit befinden. Er identifiziert fünf unterschiedliche Kategorien, die sich vor allem dadurch herausbilden, in welchem Umfang die Unternehmen Kollaborationsstrategien für sich a-

daptiert haben. Die Kategorien reichen von einem Workgroups Fokus, in dem die Unternehmen für sich keinen oder einen sehr geringen Wert in Kollaboration und Kollaborationsintegration in Geschäftsprozessen erkannt haben, bis hin zu einer Kollaborationsstrategie, die das gesamte Unternehmen durchdringt. Tabelle 1 gibt einen Überblick über die fünf von Cain identifizierten Kategorien und gibt deren wichtigste Merkmale wider. Der Nutzen für Unternehmen in der Analyse ihrer „kollaborativen Reife“ ergibt sich in zweierlei Hinsicht: „1) enterprises can gauge how they rank compared to competitors and other Global 2000 (G2000) organizations in collaboration deployments; and 2) companies gain an understanding of the path and sequence of Steps needed to progress to collaboration maturity (e.g. identify investments and remediate gaps)“. [Cain 2003, S.2]

Zu der Kategorie „Process-centric“ gehören solche Unternehmen, die begonnen haben kollaborative Services in ihre Business-Prozesse einzubinden. Diese Unternehmen haben ihre Kollaborationsstrategie auf kontextuelle Kollaboration um/eingestellt. Laut Analyse der META Group fielen 2003 unter diese Kategorie weniger als 5% der Global 2000 Unternehmen. Zu der vierten Kategorie „Workplace-centric“ gehörte 2003 noch kein Unternehmen. Das heißt, die Zielgruppe für kontextuelle Kollaboration sind vor allem die Unternehmen, die zur Kategorie „Enterprise“ gehören; das sind nach Schätzung von META-Group etwa 40% der Global 2000 Unternehmen; und die Unternehmen, die gerade dabei sind das Konzept der kontextuellen Kollaboration umzusetzen, also Unternehmen der Kategorie 3 „Process-centric“.

Zu Unternehmen der Kategorie “Workgroup” (etwa 15 % der Global 2000 Unternehmen) sagt Cain „Enterprises at this maturity level have yet to introduce second-generation collaboration technology due to perceived lack of demand or institutional stubbornness.“ [Cain 2003, S.2] Sie bieten in ihren Unternehmen keine Strukturen für die Integration von Kollaboration in die Geschäftsprozesse. Damit sind sie zurzeit für die kontextuelle Kollaboration noch nicht zugänglich.

Tabelle 1: Collaboration Categories [Cain 2003 S.4]

Maturity Level	Strategic Inflection Point	Technology	Critical Behaviors
0. Workgroup	Users still using basic collaboration systems established in the 1990s	E-mail and calendaring, simple discussion and document databases.	Lack of demand or acknowledgement of potential for next-generation collaboration to affect efficiencies.
1. Business unit	Business units implementing newer collaboration services without any deeper corporate changes.	Teamware, Web conferencing (from hosted suppliers), public IM network use.	Significant end-user demand drives acquisition without IT group oversight. Organic recognition of collaboration values.
2. Enterprise	IT groups drive acquisition and management of collaboration services for internal use on a piecemeal basis. Beginning of structural changes.	Web conferencing (on premises or hosted), private IM (or controlled public use), teamware, customer-facing chat, page push, co-browse, mail response management.	IT groups responds to demand or front-runs demand, and implements, but often independent of a larger collaboration strategy, and only tacit recognition of collaboration values.
3. Process-centric	IT groups deliver services to all corporate stakeholders and work to identify and remedy process gaps.	Generally Web conferencing, IM, teamware, and aforementioned external collaboration services.	IT groups acquire and deploy with an emphasis on reusing infrastructure and a shared-service model and work with businesspeople to create process efficiencies.
4. Workplace-centric	IT groups proactively work with business people to change the corporate culture to emphasize and reward collaborative behavior.	E-mail, calendaring, IM, presence information, Web conferencing, teamware, and other emerging and older collaboration tools.	Companies recognize the value of contextual collaboration and take steps to strategically infuse it across the extended enterprise.

2. Grundlagen

Cain ist der Auffassung, dass es wichtig ist eine Langzeitstrategie zu entwerfen, entlang derer die Integration von kontextueller Kollaboration in Teilschritten vollzogen werden kann. „To prevent IT anarchy, we recommend organizations develop a long-term strategy for making collaboration investments.” [Cain 2003, S.1] Das soll aber nicht bedeuten, dass die Kollaboration in jeder Geschäftseinheit für sich umgesetzt wird, sondern es ist wichtig dass sich überhaupt eine unternehmensweite Kollaborationsstrategie herausbildet.

Eines der spezifischen Merkmale kontextueller Kollaboration ist eine Infrastruktur, in der grundsätzlich alle Ressourcen zur gemeinsamen Benutzung bereitgestellt werden. Dadurch wird eine Umgebung geschaffen, in der der gemeinsamen Nutzung von Wissen eine besondere Bedeutung zukommt. “Another core requirement for contextual collaboration is a freely flowing network of intelligence and information.” [Chen et al. 2004, S.31]

Eine wichtige Voraussetzung, damit Mitarbeiter bereit sind ihr Wissen zu teilen besteht darin, eine dahingehende Unternehmenskultur auszubilden. [vgl. Chen et al. 2004] Vorangetrieben werden kann dieser Prozess durch mehrere Maßnahmen: Erstens ist es wichtig, Vertrauen unter Teammitgliedern zu schaffen. Für verteilte Teams kann es daher beim Start eines neuen Projekts hilfreich sein, ein Face to Face Kickoff Meeting abzuhalten. „Hopefully, the familiarity of having personally met other team Members will help them to connect with one another more often and in an effective manner, and members should be trained to respond to requests for interaction by their peers using the technological tools available.” [Chen et al. 2004, S.31]

Des Weiteren sollten standardisierte Kommunikations-Methoden unternehmensweit eingeführt werden. Darin sollten die Mitarbeiter und neue Teammitglieder regelmäßig geschult werden.

Die Unternehmensleitung kann Anreize liefern, die diese Unternehmenskultur der gemeinsamen Nutzung von Wissen und Informationen entstehen lässt. „ Most of what companies need to do to improve collaboration is not technology related. Aligning Infrastructure is one piece, but enterprises cannot „make“ people collaborate unless they look at performance management, rewards, incentives, etc.” [Cain 2003 S.4] D.h. Cain sieht es als Möglichkeit, Mitarbeiter, die dem Paradigma des „Information-sharing“ in besonderer Weise Folge leisten durch Auszeichnungen oder monetäre Leistungen zu belohnen.

Es ist wichtig, dass kontextuelle Kollaboration nicht als statisches System betrachtet wird. Es sollte ständig auf den Nutzen für die zu erledigenden Business-Prozesse überdacht und überprüft werden. Es sollten Verbesserungsvorschläge von den Teammitgliedern ernst genommen werden, da sie diejenigen sind, die in dieser Umgebung täglich kollaborieren. Das ist besonders wichtig für die verteilten Teams, die auf elektronische Kollaborationsmittel angewiesen sind um ein positives Prozessergebnis zu erzielen.

2.4 Grundlagen zur Portalarchitektur

Webbasierte Portale haben sich in den letzten Jahren neben ihrer Funktion als Integrationsplattform auch immer mehr als geeignete Architektur für unternehmensinterne kollaborative elektronische Arbeitsplätze (e-Workplaces) für Knowledge-Worker herauskristallisiert. Die in Kapitel 2.3 hergeleitete Definition von kontextueller Kollaboration stellt die Integration von kollaborativen Funktionalitäten als eines der Kennzeichen kontextueller Kollaboration heraus. Portale als Integrationsplattform scheinen daher besonders geeignet kontextuelle Kollaborationsarchitekturen zu realisieren. Auch die im vierten Kapitel evaluierte *IBM Workplace* Architektur zum Beispiel ist Portal basiert. Sie baut auf der *IBM WebSphere Portal*-Architektur auf.

Ein Portal, also ein Eingang, wird in seiner metaphorischen Bedeutung sehr anschaulich von Oesterle et al. definiert: „Ein Portal ist-technisch gesprochen- das Fenster zu allen Funktionalitäten aus unterschiedlichen Applikationen.“ [Oesterle et al. 2002, S.23] „Portale können als webbasierte personalisierbare und integrierte Zugangssysteme zu Content, Applikationen und Services für einen bestimmten Anwendungszweck verstanden werden.“ [Oesterle et al. 2002, S.97]

Basisgebende strukturelle Gemeinsamkeiten von verschiedenen am Markt verfügbaren Portalen wurden von [Hahnl 2004] identifiziert. Die Abb. 5 zeigt diese grundlegende Struktur.

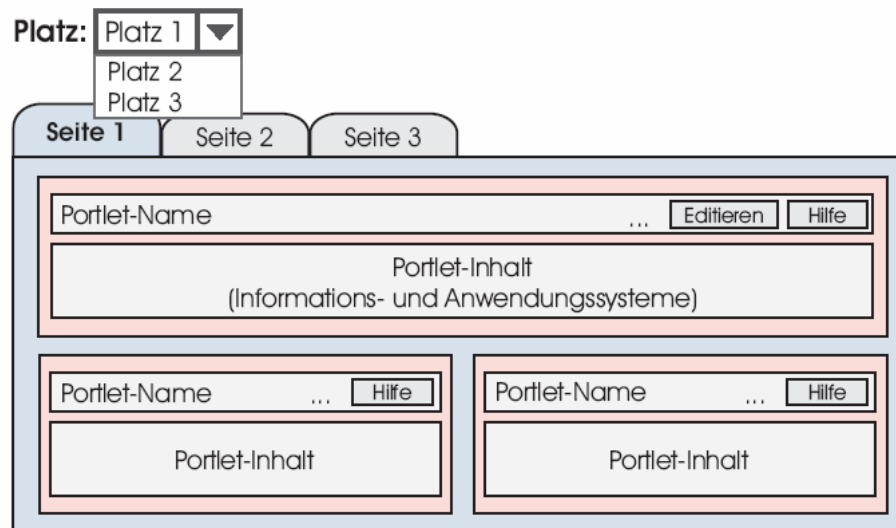


Abb. 5: Gemeinsame Strukturelemente verschiedener Portalimplementierungen

[Hahnl 2004 S.84]

Als Basiskomponenten nennt Hahnl *Informations- und Anwendungssysteme*, *Portlets*, *Seiten*, und *Plätze*.

Basisdienste von Portalen sind z.B.: *Personalisierung*, *Suche*, und *Echtzeitkollaborationsfunktionalitäten*. [vgl. Hahnl 2004 S.84 ff] Es soll wird hier kurz auf die genannten Komponenten und Dienste eingegangen.

2.4.1 Basiskomponenten von Portalen

- **Informations- und Anwendungssysteme**

Informations- und Anwendungssysteme dienen der Verwaltung, Verarbeitung und Speicherung der unternehmensrelevanten Daten, Informationen und Prozesse. Systemkategorien sind unter anderem Enterprise Resource Planning-, Produktionsplanungs-, Produktionssteuerungs-, Groupware-Systeme sowie Desktop-Büro-Anwendungen.

- **Portlets**

Hahnl beschreibt Portlets als „kleinste Informationseinheit eines Portals“, die das Bindeglied zwischen den Informations- und Anwendungssystemen und den anderen Portaldiensten sind. Weiter heißt es: „Sie bieten den Benutzern des Portals

den personalisierten Zugang zu Informationen, Applikationen, Prozessen und Personen, indem die anzuzeigenden Informationen aus den dahinter liegenden Informations- und Anwendungssystemen abgerufen und im Portlet dargestellt werden.“ [Hahnl 2004, S.85] Der Zugriff auf das Portlet wird durch Zuweisung von Rechten durch Administratoren geregelt. Dadurch und durch Anpassungsfähigkeit (customizing) der Portlets bekommt der Benutzer eine individuelle Sicht auf den Informationsbestand. Die Anzeige kann auf unterschiedlichen Ausgabegeräten erfolgen, wie Web-Browser, Mobiltelefon, Handheld etc. Portlets dienen nicht nur der Anzeige, sondern dienen auch als Benutzerschnittstelle zur Durchführung komplexer Prozesse und Aufgaben. [vgl. Hahnl 2004, S.85]. Der Begriff “Portlet” wird dabei nicht einheitlich von allen Herstellern genutzt. „The Term Portlet is used by IBM Oracle Bea and Sun, Other vendors use terms like gadgets and Web parts.” [Iyengar et al. 2005, S.6])

- **Seiten**

Portalseiten können als Container für die einzelnen Portlets angesehen werden. Das heißt, mehrere Portlets werden auf einer Seite gruppiert. „Die Seite bringen die vormals voneinander unabhängigen Portlets und damit die Informations- und Anwendungssysteme in einen neuen Kontext.“ [Hahnl 2004, S.86]

- **Plätze**

„Plätze (engl. places) sind die höchste Aggregationsebene eines Portals. Sie dienen der weiteren Strukturierung und fassen zusammengehörige Seiten, z.B. basierend auf Projekten, Aufgaben oder Organisationseinheiten, zusammen.“ [Hahnl 2004, S.87] Durch Plätze werden Teamumgebungen für lokale verteilte oder virtuelle Teams geschaffen. Hahnl identifiziert zwei grundsätzliche Arten von Plätzen:

Persönliche Plätze (Private Places)

Jeweils nur ein Anwender hat Zugriff.

Es stehen ihm ein oder mehrere persönliche Plätze zur Verfügung. Sie dienen der Organisation der persönlichen Arbeitsumgebung

Gemeinsam genutzte Plätze (Shared Places)

Mehrere Anwender haben auf den gleichen Platz Zugriff.

Der Platz kann durch Zugriffskontrolle für eine definierte Personengruppe beschränkt sein oder er kann vollständig offen sein.

2.4.2 Basisdienste von Portalen

Die drei von Hahnl genannten Dienste sind eine Auswahl aus zahlreichen von Portalanbietern implementierten Diensten. Diese Dienste sind aber auch für Portal-Architekturen, die unter dem Aspekt der kontextuellen Kollaboration realisiert werden von Bedeutung und werden deshalb kurz erläutert.

- **Personalisierung**

Die Personalisierung macht es möglich, sowohl die abrufbaren Informationen als auch die Auswahl und Anzeige der Portlets für jeden einzelnen Benutzer zu individualisieren. Die Personalisierung wird in der Regel durch Benutzerprofile realisiert. „Die Grundlage von Personalisierungsfunktionen sind Profile, die Informationen über Rollen, Aufgaben, Interessen, Nutzungsverhalten und weitere auswertbare Aspekte des Benutzers enthalten.“ [Hahnl 2004, S.90]

- **Suche**

Suchfunktionalität ermöglicht dem Benutzer über die gesamte Breite der ihm im Portal zur Verfügung stehenden Daten zu suchen, unabhängig davon wie die Informationen in den einzelnen Portlets organisiert sind. Die Suche erfolgt durch im Portal integrierte oder externe Suchmaschinen.

- **Echtzeitkollaborationsfunktionalitäten**

Hahnl führt aus, dass in den meisten am Markt zur Verfügung stehenden Portalen Kollaborationsfunktionalitäten [Kollaborationsfunktionalitäten vgl. Kapitel 2.1] integriert sind, um „Kommunikation und Zusammenarbeit [...] zwischen den verteilten Mitarbeitern [zu] verbessern und gleichzeitig zur Kostensenkung bei[zu]tragen, z.B. durch Reduktion der notwendigen Reisen.“ [Hahnl 2004, S.91]

Hahnl merkt an, dass es auf Grund fehlender Standards selten oder sehr schwer möglich ist, andere Produkte die bereits innerhalb eines Unternehmens zur Kollaboration genutzt werden, in die Portale zu integrieren. Es soll im Rahmen dieser Arbeit ein Framework vorgestellt werden, in der die Integration anderer

Komponenten vorgesehen ist da Integrationsfähigkeit eine wichtige Voraussetzung darstellt um eine kontextuell-kollaborative Architektur zu realisieren.
[Hahnl 2004, S.91]

3. Anforderungen und Konzepte eines Frameworks zur kontextuellen Kollaboration

Dieses Kapitel beschreibt die Anforderungen und Konzepte eines Frameworks zur kontextuellen Kollaboration. Zunächst wird in Kapitel 3.1 ein beispielhaftes Business-Szenario vorgestellt, welches die verschiedenen Aspekte kontextueller Kollaboration in der Praxis verdeutlicht. Das Kapitel 3.2 stellt dann das im Rahmen dieser Arbeit entwickelte **vierstufige Modell der kontextuellen Kollaboration** vor. Dieses Modell wird im Folgenden zur Strukturierung der Anforderungen an ein idealtypisches Framework zur kontextuellen Kollaboration sowie zur Herleitung funktionaler und struktureller Systemmerkmale herangezogen, die diese Anforderungen umsetzen.

3.1 Anwendungsszenario: Kontextuelle Kollaboration in einem Logistikunternehmen

Zunächst wird ein fiktives aber praxisnahes Business-Szenario geschildert in dessen Kontext verschiedene Use-Cases betrachtet werden, die jeweils verschiedene Aspekte kontextueller Kollaboration beleuchten. Diese Use-Cases beschreiben, wie verschiedene Personen aus dem Kontext des gegebenen Business-Szenarios miteinander kollaborieren. Sie können gemeinsame Dokumente und andere Artefakte erstellen und bearbeiten – jeweils direkt aus dem Umfeld ihrer derzeitigen Arbeit erstellen und bearbeiten, ohne den Kontext des Geschäftsprozesses, den sie gerade bearbeiten verlassen zu müssen.

Diese Use-Cases wurden entsprechend der Methode des Storyboarding entworfen [vgl. Gruen 2000]. Gruen beschreibt diese Methode als geeignet zur Spezifikation und Verdeutlichung von Anforderungen. „Because stories capture in detail the real-world context in which a new technology will be used, they help determine which functions will be useful, how they should be presented, and what integration with other tools, people and information will be important.” [Gruen 2000, S.1]

3.1.1 Das Szenario

InfraTrans ist ein international agierendes Logistikunternehmen mit 56 Niederlassungen in verschiedenen, zumeist europäischen Ländern aber auch in Nord- und Südamerika. Alice arbeitet als Projektleiterin in der deutschen Niederlassung von *InfraTrans*. Sie ist dafür verantwortlich die Logistikprojekte großer Kunden von Anfang bis Ende zu begleiten und zu organisieren. Beth ist Alices Managerin. Sie ist die Abteilungsleiterin für

Großkundenprojekte und hat Alice ein neues Projekt zugewiesen. *Scherer Chemicals Ltd.*, einer der größten Kunden von *InfraTrans*, hat eine Anfrage für eine Angebotserstellung für den Transport von 12.000 Litern eines flammhemmenden Zusatzstoffes für Speziallacke an einen Abnehmer in der französischen Schweiz gestellt. Obwohl eine Angebotserstellung ein alltäglicher Vorgang für Alice ist, werden spezielle rechtliche Überlegungen und Maßnahmen erforderlich, um die Einstufung des Zusatzstoffes in eine Gefahrgutklasse sowie seinen Transport in ein Land außerhalb der EU möglich zu machen.

3.1.2 People-Awareness im Kontext einer CRM-Applikation

Da Alice erst eineinhalb Monate in der Abteilung für Großkunden arbeitet, ist dies ihr erstes Projekt mit *Scherer Chemicals*. Alice will mehr Informationen über diesen Kunden herausfinden und öffnet das unternehmensinterne Sales-Portal von *InfraTrans*. In diesem Sales-Portal findet sie auch Zugang zu dem internen Customer-Relationship-Management (CRM) System, in dem *InfraTrans* alle relevanten Daten zu ihren Kunden pflegt, inklusive Informationen zu Kontaktpersonen, vorherigen Projekten und zur bisheriger Korrespondenz mit diesem Kunden. Alice betrachtet die Liste der bisherigen Projekte die für *Scherer Chemicals* in der Vergangenheit durchgeführt wurden und sieht dabei, dass Dave bis zu ihrem Wechsel in die Abteilung „Großkunden“ für *Scherer* verantwortlich war. Alice findet ein Projekt aus dem Jahr 2005 welches ganz ähnliche Rahmenbedingungen hatte wie das derzeitige. Auch dieses Projekt hatte Dave geleitet. Durch die in das CRM-Portal integrierte Awareness kann Alice sehen, dass Dave gerade online ist, allerdings nicht gestört werden möchte. Dies wird durch das Status-Icon deutlich gemacht welches vor Daves Namen im CRM-Portlet sichtbar ist.

3.1.3 Senden einer E-Mail aus dem Kontext einer CRM-Applikation

Alice beschließt Dave eine E-Mail zu schicken um ihn zu näheren Details des Projektes aus dem vergangenen Jahr zu befragen. Sie klickt in der Projektübersicht auf Daves Namen und öffnet damit ein Kontextmenü, welches neben einigen Informationen über Dave wie z.B. seinem Bild sowie seiner Telefonnummer und E-Mail Adresse auch Zugang zu verschiedenen Kollaborationsfunktionen zur Verfügung stellt, wie z.B. „Sende E-Mail“, „Sende Instant Message“, „Rufe an“ und „Starte e-Meeting“. Alice klickt auf „Sende E-Mail“, wodurch sich für Alice eine neue E-Mail zur Bearbeitung öffnet. In dieser E-Mail wurde bereits Daves E-Mail Adresse als Empfänger eingetragen. Da die

E-Mail aus dem Kontext der Projektdetailansicht für *Scherer* gestartet wurde, wurde auch die Betreffzeile mit dem Titel des Projektes „*Scherer Chemicals* Projekt 24/2005“ belegt, sowie automatisch ein URL-Link auf diese Projektansicht in das Bodyfeld der E-Mail kopiert. Sie ergänzt die Mail mit einigen Angaben der aktuellen Angebotsanfrage von *Scherer* und fragt Dave, ob er ihr vielleicht bei der Erstellung eines Angebotes für diesen Fall helfen könne.

Eine halbe Stunde später bekommt Alice eine Instant Message von Dave der gerade sein Telefonat mit einem Kunden beendet und nun Alices E-Mail gelesen hat. Dave bietet Alice an, ihr bei der Angebotserstellung zu helfen. Allerdings sei auch er kein Experte für Gefahrguttransporte in die französische Schweiz, doch habe er schon häufiger mit Pierre aus der *InfraTrans* Niederlassung in Genf zusammengearbeitet. Pierre sei Experte für Gefahrguttransporte in die Schweiz und könne sicherlich bezüglich der notwendigen Bestimmungen helfen. Als Ausgangspunkt für ein Angebot könne Alice auf das Angebot für das Projekt 24/2005 für *Scherer* zurückgreifen, welches Dave letztes Jahr erstellt hat.

3.1.4 Anlegen einer Activity zur ad-hoc Koordination einer Angebotserstellung

Alice hält dies für eine sehr gute Idee und startet direkt eine neue Activity, um die Erstellung des Angebotes für *Scherer* in Kooperation mit Dave und Pierre zu koordinieren. Dazu öffnet sie den Menüpunkt „Activities“ in der Menüzeile ihres Instant-Messaging-Fensters. Darunter findet sie die Einträge „Neue Activity starten“ und „Zu bestehender Activity hinzufügen“. Alice klickt auf „Neue Activity starten“ wodurch direkt eine neue Activity angelegt wird und eine Datei mit dem Transkript des Chats als Inhalt in die Activity gelegt wird. Zudem werden Alice und Dave als die beiden Instant-Messaging-Teilnehmer direkt als Mitglieder der Activity hinzugefügt. Alice gibt der Activity den Namen „Angebotserstellung *Scherer* Februar 2006“ und braucht nun nur noch Pierre zu der Activity hinzufügen. Dies erreicht sie durch das Klicken auf Pierres Namen in dem Chat-Transkript mit Dave, welches ein Kontextmenü öffnet, in dem neben den üblichen Funktionen zur direkten Kollaboration nun auch ein Eintrag „Zu bestehender Activity hinzufügen“ verfügbar ist. Alice legt nun noch ein Dokument in der Activity an, in der sie die Anforderungen und Rahmenbedingungen des neuen Projektes mit *Scherer* im Detail beschreibt, damit Dave und Pierre die notwendige Hintergrundinformation zu diesem Geschäftsvorgang haben.

3.1.5 Kontextuelle Integration einer CRM-Applikation und einer Dokumentbibliothek

Alice will nun einen ersten Entwurf eines Angebotes für *Scherer* auf Basis des alten Angebotes von Dave erstellen. Hierzu geht sie zurück in das Sales-Portal auf die Seite mit dem CRM-Portlet. Neben dem CRM-Portlet findet sie auf der gleichen Seite auch eine Dokumentenbibliothek-Komponente. Diese verteilte Dokumentenbibliothek beinhaltet alle Dokumente mit Angeboten und Anfragen, die für einen Kunden erstellt wurden. Die Dokumenten-Komponente ist dabei mit dem CRM-Portlet verbunden. Wenn der Benutzer einen Kunden aus der Kundenlistenansicht im CRM-Portlet auswählt, wird diese Kontextinformation über den ausgewählten Kunden an das Dokumentenbibliothek-Portlet übertragen. Dort wird es als Filterkriterium für die enthaltenen Dokumente benutzt. Das Dokumentenbibliothek-Portlet zeigt dann nur die Dokumente an, die für den ausgewählten Kunden erstellt wurden. Alice wählt *Scherer Chemicals* aus der Kundenliste aus und bekommt darauf hin alle Dokumente in der Dokumentenbibliothek angezeigt, die für *Scherer* angelegt wurden. Unter anderem findet Alice hier auch das Angebot für das Projekt „*Scherer 24/2005*“ das Dave erwähnt hatte.

3.1.6 Zufügen eines Dokumentes zum Kontext einer Activity

Sie kopiert das Dokument, öffnet es und schaut sich das Angebot an. In dem damaligen Angebot ging es um einen Stoff einer anderen Gefahrgutklasse und auch nur um ca. ein Drittel der Menge für das aktuelle Angebot. Daher rechnet Alice die Kosten hoch, setzt die neue Gefahrgutklasse in das Angebot ein und korrigiert den Gesamtpreis sowie den Titel und das Datum für das neue Angebot. Dann will sie es Dave und Pierre zur Begutachtung und Kommentierung vorlegen. Dazu findet sie im Menü ihrer Textverarbeitung neben dem Menüpunkt „Speichern“ und „Speichern unter...“ auch einen Eintrag „Speichern in Activity...“. Nachdem sie diesen Punkt angeklickt hat, öffnet sich ein Auswahldialog, in dem sie alle Activities angezeigt bekommt, in denen sie gerade Mitglied ist – sortiert danach, welche Activities sie zuletzt bearbeitet hat. Sie findet ihre Activity „Angebotserstellung *Scherer* Februar 2006“ ganz oben in der Liste, wählt diese aus und klickt auf „OK“, wodurch das Dokument der Activity hinzugefügt wird. Anschließend kopiert sie noch einen URL-Link auf den Kundendatensatz von *Scherer* aus dem CRM-Portlet in die Activity und schließt dann die Activity.

3.1.7 Erstellen eines Benachrichtigungsprofils für eine Activity

Da sie das Angebot besonders schnell fertig stellen und an *Scherer* zurücksenden will, möchte sie sich sofort benachrichtigen lassen, falls Dave oder Pierre irgendwelche Änderungen an oder Ergänzungen zu den Dokumenten in dieser Activity machen. Sie klickt dazu auf die Activity in ihrer Activityliste und wählt den Menüpunkt „Benachrichtigen wenn...“ aus dem Kontextmenü der Activity aus. In dem darauf folgenden Dialog kann sie die Benachrichtigungsoptionen auswählen. Sie klickt an, dass sie über jegliche Änderungen an der Activity benachrichtigt werden möchte und zwar sowohl durch eine E-Mail-Benachrichtigung als auch durch eine Instant Message.

Einige Zeit später kommt Pierre in der Niederlassung Genf an seinen Arbeitsplatz zurück und meldet sich beim System an. Direkt nach dem Einloggen bekommt er die Benachrichtigung, dass eine neue Activity angelegt wurde, in der er als Mitglied eingetragen wurde. Pierre öffnet die Activity und sieht aus dem Chat-Transkript und den verschiedenen Dokumenten, dass Alice ein Angebot über einen Gefahrguttransport in die Schweiz für *Scherer Chemicals* erstellen möchte. Er schaut sich den Angebotsentwurf an, den Alice in die Activity gestellt hat und stellt fest, dass es sich um einen Transport der Gefahrgutklasse 6.1 handelt. Gefahrguttransporte dieser Klasse dürfen in der Schweiz aber auf einigen Strassen und Autobahnen nicht transportiert werden und müssen dann streckenweise per Bahn transportiert werden. Da Alice diesen Aspekt nicht berücksichtigt hatte und der Transport per Bahn ca. 30 Prozent teurer ist als über die Strasse, schreibt Pierre eine Notiz mit seinen Änderungsvorschlägen und hängt sie als Antwort an das Angebotsdokument in der Activity. Außerdem fügt er noch einen Link auf ein Dokument zur Activity hinzu, welches die Richtlinien für Gefahrguttransporte in die Schweiz beschreibt. Dieses Dokument liegt in einer unternehmensweit verteilten Dokumentenbibliothek für Richtliniendokumente.

Nachdem Pierre die Activity schließt, wird Alice direkt über die Änderungen vom System benachrichtigt. Alice befindet sich gerade in ihrem Mail-Client um die E-Mails des Tages zu bearbeiten, als eine Instant Message sie darüber informiert, dass Pierre neue Dateien in die Activity gestellt hat. Durch einen Klick auf den Link in der Instant Message öffnet sich die Activity. Alice sieht Pierres Antwort auf ihr Angebotsdokument und liest seine Kommentare dazu. Sie sichtet noch das Richtliniendokument das Pierre referenziert hat und übernimmt dann Pierres Änderungsvorschläge in ihr Angebot.

Jetzt wo das Angebot vollständig ist, schickt sie es per Mail an *Scherer Chemicals*. Dazu klickt sie auf das Angebotsdokument und wählt den Befehl „Sende Dokument..“ aus dem Kontextmenü. Dadurch öffnet sich eine neue E-Mail, in der das Angebot schon als Dateianhang eingefügt ist sowie der Titel des Dokumentes als Mail Subjekt übernommen wurde. Alice fügt nur noch den Adressaten sowie ein passendes Anschreiben dazu und verschickt die E-Mail dann per „Senden und Speichern“. Dadurch wird die E-Mail versandt und zusätzlich eine Kopie der Mail in die Activity eingefügt.

3.1.8 Erstellen eines Teamspace zur Durchführung des Projektes

Drei Tage später bekommt Alice die Mitteilung, dass *Scherer Chemicals* das Angebot angenommen und den Auftrag erteilt hat. Damit kann sie nun anfangen, das Projekt zu planen und durchzuführen. Der Transport des Gefahrgutes muss in 3 Wochen beendet sein. Neben Dave und Pierre, die sie auch weiter bei diesem Projekt beratend unterstützen, arbeiten noch zwei Kollegen aus ihrem Team an diesem Projekt mit – Bob, der für die vorbereitenden Arbeiten bei *Scherer* in Hamburg zuständig ist, sowie Lisa, die alle Aufgaben für die Übergabe an den Abnehmer in der Schweiz erledigt. Die Durchführung eines solchen Logistikprojektes bedarf verschiedener Vorarbeiten: Die Ausarbeitung eines Projektvertrages; das Erstellen eines genauen Zeitplanes; die Erledigung der Formalien, die mit dem Gefahrguttransport verbunden sind, wie z. B. die Deklaration und Anmeldung des Gefahrgutes bei den deutschen und schweizerischen Behörden usw. Um die Arbeit dieses Teams zu koordinieren und jedem Teammitglied Zugriff auf die mit diesem Projekt zusammenhängenden Dokumente und Informationen zu geben, erstellt Alice einen gemeinsam genutzten Teamspace („Shared-Teamspace“) für dieses Projekt. Dieser Teamspace basiert auf einem von Alice erweiterten Template eines Projekt-Teamspace, welches Alice schon erfolgreich in mehreren Projekten benutzt und erweitert hat.

3.1.9 Der Shared-Teamspace als „Composite-Application“

Der Teamspace beinhaltet verschiedene Komponenten die für die kollaborative Projektbearbeitung im Team benötigt werden: eine Dokumentenbibliothek zur Speicherung und Verwaltung aller für dieses Projekt erstellten Dokumente und Formulare, einen Teamkalender um gemeinsame Termine und Projektmeilensteine für das Team zu koordinieren, eine Projektmanagement Komponente, um die einzelnen Aufgaben der Teammitglieder in einem Projektplan zu managen und deren Bearbeitungsfortschritt

festzuhalten und in einem Gantt Chart anzuzeigen, ein Diskussionsforum, um aufkommende Fragen und Probleme für das gesamte Team zu diskutieren und klären, eine CRM-Komponente, die den Zugriff auf das zentrale CRM-System der *InfraTrans* und damit alle Kundendaten zu *Scherer Chemicals* ermöglicht sowie eine von *InfraTrans* selbst entwickelte Komponente zur Belegungsplanung und Verwaltung der LKW-Flotte sowie der zur Verfügung stehenden Fahrer an den verschiedenen *InfraTrans* Standorten. Zusätzlich beinhaltet der Teamspace noch eine Mitglieder-Komponente, die die Liste aller zu diesem Teamspace gehörenden Mitglieder sowie deren Rollen in diesem Projekt anzeigt. Alice bearbeitet die CRM-Komponente, indem sie im Konfigurationsdialog dieser Komponente „*Scherer Chemicals*“ als Filterkriterium definiert. Dadurch zeigt die CRM-Komponente in diesem Teamspace nur noch Informationen die im Zusammenhang mit dem ausgewählten Kunden stehen. Alice gibt diesem neuen Teamspace den Titel „*Scherer Chemicals* Projekt 01/2006“ und trägt alle an diesem Projekt beteiligten Personen als Mitglieder in diesen Teamspace ein. Sie werden automatisch durch eine E-Mail darüber informiert, dass dieser Projekt-Teamspace neu erstellt wurde.

3.1.10 Kontextuelle Kollaboration im Teamspace

Einige Zeit später sieht Bob die E-Mail-Benachrichtigung in seinem E-Mail-Eingangskorb. Er klickt auf den enthaltenen Link und wird zu dem neu angelegten Teamspace gebracht. Vorher fordert ihn das System allerdings auf, sich durch ein Login anzumelden, wodurch er automatisch zu allen Anwendungen und Komponenten angemeldet wird, auf die Bob Zugriff hat. Im Teamspace sieht Bob, dass Alice schon einige Dokumente für dieses Projekt in die Dokumentenbibliothek innerhalb des Teamspaces kopiert hat. Diese stammen aus der Activity, die Alice zur Erstellung des Angebotes gestartet hatte. Bob muss nun zuerst die Gefahrgutanmeldung für die deutschen Behörden erstellen, da das Genehmigungsverfahren für diese Gefahrgutklasse normalerweise einige Tage dauert. Bob weiß, dass Lisa schon viele Projekte mit Gefahrguttransporten abgewickelt hat und hofft, von ihr vielleicht ein Beispiel für eine solche Gefahrgutanmeldung als Ausgangspunkt bekommen zu können. Durch die Place-based-Awareness-Funktionalität der Mitglieder-Komponente sieht Bob, dass Lisa online ist und sich gerade auch in dem Teamspace aufhält. Er nutzt die Gelegenheit und startet einen Chat mit Lisa. Lisa erzählt ihm, dass sie erst vor zwei Monaten ein ähnliches Projekt mit der gleichen Gefahrgutklasse bearbeitet hat. Bob öffnet daraufhin die Such-Funktion in diesem Teamspace und tippt den Suchbegriff „Gefahrgutklasse 6.1“ in den Dialog. Zusätz-

lich hat er noch die Optionen „In diesem Teamspace suchen“ und „In allen Teamspaces suchen“ von denen er die letztere Option auswählt. Nach kurzer Zeit bekommt er eine Reihe von Treffern präsentiert, unter denen sich auch Lisas Gefahrgutanmeldung von vor zwei Monaten befindet. Sie liegt in einem Teamspace, der für ein Projekt mit der *Bennings Verfahrenstechnik GmbH* erstellt wurde. Dieser Teamspace ist mittlerweile archiviert und erlaubt nur noch Lesezugriff auf die darin enthaltenen Dokumente. Bob kopiert sich das Dokument als Vorlage in den neuen Teamspace und passt die darin enthaltenen Daten an das neue Projekt an.

3.1.11 Navigation zwischen verschiedenen Kontexten: Projekt, Kunde, Teamspace

Zur gleichen Zeit arbeitet Alice an der Fahrzeug- und Fahrerplanung für den Transport. Alice möchte einen Fahrer einsetzen der möglichst schon einmal für *Scherer* gefahren ist und deshalb ortskundig und mit dem Kunden vertraut ist. Dazu sucht Alice erstmal alle Projekte der letzten Zeit, die für *Scherer* durchgeführt wurden und sucht dann die Liste der Fahrer, die für die Projekte zugeteilt worden waren. Dazu geht Alice auf den Kundendatensatz von *Scherer* in der CRM-Komponente und öffnet aus dem Kontextmenü den Navigationsdialog für das Objekt „*Scherer Chemicals*“. In diesem Dialog befindet sich eine Liste von Objekttypen die mit dem gewählten Kunden in Beziehung stehen. Alice wählt „Zeige alle Teamspaces“ aus der Liste, klickt OK und bekommt eine Menge von 30 Projekt-Teamspaces, die für *Scherer* in den letzten zwei Jahren angelegt wurden. Alice schaut durch die Liste und sieht, dass sechs von den 30 Projekten auch Gefahrguttransporte waren. In fünf von diesen sechs Fällen war Robert als Fahrer eingeteilt. Alice erinnert sich, dass Dave Robert als zuverlässigen Fahrer erwähnt hat und beschließt daher, Robert als Fahrer für ihr aktuelles Projekt einzuteilen. Sie öffnet ihren aktuellen Teamspace und geht zu der darin enthaltenen Komponente zur LKW- und Fahrer-Belegungsplanung. Als sie versucht Robert für die drei Tage im März einzuteilen, stellt sie fest, dass er in dieser Zeit schon für das Projekt „*Braun Automobilteile AG 03/2006*“ gebucht worden ist. Wieder benutzt Alice die Kontextnavigation, um aus der Belegungsplanungskomponente durch einen Rechtsklick auf den Projektnamen den dazugehörigen Projekt-Teamspace zu öffnen. Alice liest sich in dem Teamspace die Details zu dem Projekt für Braun Automobilteile durch und stellt fest, dass es sich hierbei um keinen Gefahrguttransport handelt. Außerdem sieht sie in dem Teamspace, dass Chris der Projektleiter dieses Projektes ist. Chris ist ein ehemaliger Kollege aus Alices

alter Abteilung. Sie beschließt daher, ihn zu fragen, ob sie Robert für ihren Gefahrguttransport einplanen kann und Chris vielleicht einen anderen Fahrer einteilen kann.

3.1.12 Kollaboration und Device-based-Awareness

Alice sieht an dem Status-Icon vor Chris' Namen, dass er gerade online ist, allerdings nicht an seinem Arbeitsplatz sitzt, sondern nur mit seinem Mobiltelefon erreichbar ist. Da Alice die Belegungsplanung schnell abschließen möchte klickt sie auf seinen Namen und erhält ein Kontextmenü, aus dem sie verschiedene Kommunikationsfunktionen starten kann. Da Chris gerade nur an seinem Mobiltelefon angemeldet ist, fehlen in diesem Kontextmenü manche Optionen wie „Starte Chat“ oder „Starte E-Meeting“, dafür kann Alice andere, device-spezifische Kommunikationsfunktionen starten wie z.B. „Sende SMS“. Also wählt sie diese Option und sendet Chris eine Kurznachricht, in der sie ihr Anliegen schildert. Chris sendet ihr eine SMS-Nachricht zurück, in der er erklärt dass er sich gerade bei einem Kunden befindet und die Sache erledigen wird, sobald er wieder zurück im Büro ist. Diese Nachricht sieht Alice als Instant Message.

3.1.13 Gemeinsames Nutzen eines Dokumentes im Kontext des Teamspace

Einige Tage später bekommt Alice einen Anruf von ihrer Managerin Beth. Beth muss einen Statusbericht über ihre Abteilung fertig stellen, den sie in zwei Tagen in einem regelmäßigen Quartals-Meeting ihrem Vorgesetzten Tom vorstellen muss. Das Projekt mit *Scherer* ist das derzeit größte Projekt in ihrer Abteilung und durch sein Volumen sehr wichtig zum Erreichen der angestrebten Umsatzzahlen für das Quartal. Daher fragt Beth Alice, ob sie den Projektfortschritt in einer kurzen Präsentation darstellen kann. Alice beginnt damit einige Folien und eine Agenda für die Präsentation zu erstellen. Sie will dann das Präsentationsdokument mit Bob gemeinsam bearbeiten, um von ihm Feedback dazu zu bekommen. Dazu stellt sie das Dokument in den Teamspace. Als Bob sich wenig später wieder am System anmeldet, bekommt er eine Benachrichtigung über das neue Dokument, welches Alice in den Teamspace gestellt hat. Als Bob die Präsentation öffnet, um sie durchzusehen sieht Alice durch ein Status-Icon an dem Dokument, dass Bob gerade aktiv an dem Dokument arbeitet. Alice nutzt die Gelegenheit und startet einen Chat mit Bob aus dem Dokument heraus. Nach einigen Wortwechseln im Chat öffnet Alice einen Audio Kanal zu Bob via Voice-over-IP (VoIP) und startet eine Screen-Sharing Sitzung aus dem Chat, um die Präsentation mit Bob detaillierter zu be-

3. Anforderungen und Konzepte

sprechen. [Storyboarding in Anlehnung an: Silva Filho et al. 2005, S.2] Alice und Bob diskutieren einige Änderungen und arbeiten sie direkt online in die Präsentation ein. Anschließend sendet Alice die Präsentation an Beth.

3.2 Das Vier-Stufenmodell der kontextuellen Kollaboration

Um ein idealtypisches Systemmodell für ein Framework zur kontextuellen Kollaboration aufzustellen, ist es wichtig zunächst die Anforderungen an ein solches System zu identifizieren. Ein genaues Verständnis der Anforderungen ist dabei essentiell um die spezifischen Systemmerkmale und Architekturprinzipien, die sich für kontextuelle Kollaboration ergeben, herzuleiten. Dazu wurden die in der Literatur und in praxisnahen Berichten beschriebenen, und durch das Szenario verdeutlichten, Eigenschaften kontextueller Kollaboration identifiziert und systematisiert.

Im Rahmen dieser Arbeit wurde daraus ein Vier-Stufenmodell der kontextuellen Kollaboration entwickelt. Dieses Modell beschreibt vier aufeinander aufbauende Integrationsstufen kollaborativer Funktionalität mit jeweils wachsender Mächtigkeit der Kontextualisierung. Diese Stufen sind:

1. Das Starten kollaborativer Funktionen aus dem Arbeitskontext
2. Das Einbinden des Arbeitskontextes in die Kollaboration
3. Die Persistierung der Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen
4. Navigation durch das multidimensionale Geflecht von Kontextbeziehungen

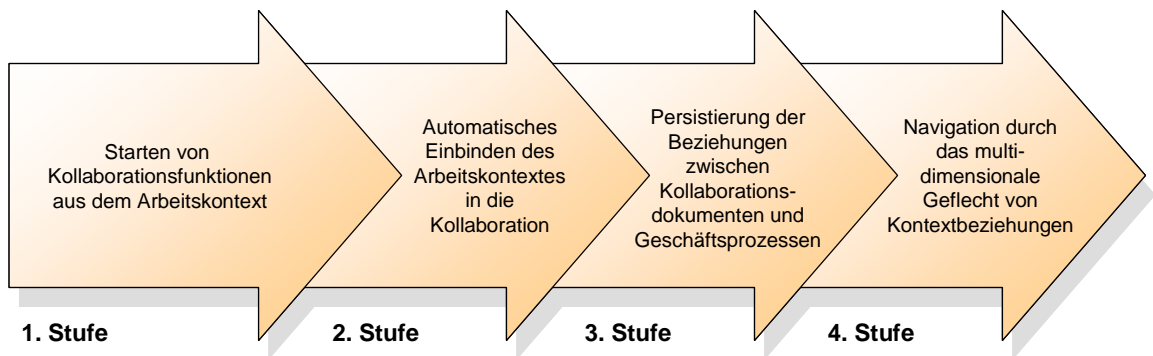


Abb. 6: Vier-Stufenmodell der kontextuellen Kollaboration

Dieses Model wird im Folgenden zur Systematisierung der Konzepte und Anforderungen an kontextuelle Kollaboration herangezogen. So werden für jede Stufe zuerst die Probleme beschrieben, die sich ohne kontextuelle Kollaboration ergeben, danach werden die Vorteile der kontextuellen Kollaboration herausgestellt sowie ein Satz von Anforderungen formuliert welche diese Probleme adressieren. Schließlich werden funktio-

nale und strukturelle Systemmerkmale aufgestellt, die genau diesen Katalog von Anforderungen adressieren und eine konkrete Implementierung eines solchen kontextuell-kollaboratives Systems ermöglichen.

Dieses Vier-Stufenmodell wird nun im Folgenden im Detail erläutert.

3.2.1 Stufe 1: Starten kollaborativer Funktionen aus dem Arbeitskontext

Kollaborative Werkzeuge wie E-Mail oder Instant Messaging stehen in den Unternehmen heute üblicherweise in Form von separaten Stand-alone-Applikationen oder monolithischen Groupware-Systemen bereit. Wenn in diesem Umfeld ein Mitarbeiter während seiner Arbeit an einem Geschäftsprozess, z.B. während der Bearbeitung einer Kundenbestellung in einem SAP-System, eine Rückfrage an einen Kollegen hat, der die Bestellung beim Kunden aufgenommen hat, so muss er das SAP-System verlassen und seinen separaten E-Mail oder Chat-Client starten, um mit dem Kollegen in Kontakt zu treten. Dies bedeutet für den Mitarbeiter, dass er bei der Benutzung der Kollaborationswerkzeuge den Bezug zu seinem ursprünglichen Arbeitskontext verliert. Neben dem Verlust des Kontexts kann dieser Wechsel z.B. in seinen E-Mail-Client auch bedeuten, dass der Mitarbeiter von seinem ursprünglichen Vorhaben abgelenkt wird weil er sich noch anderen E-Mails zuwendet, die sich in seinem E-Mail-Eingangskorb angesammelt haben.

Kontextuelle Kollaboration adressiert genau dieses Problem, indem die Kollaboration, in diesem Fall der Chat, direkt aus der Business-Applikation heraus angestoßen werden kann. „Embedding collaboration seamlessly into host applications spares users the time and effort of context switching to other tools whenever they need to communicate or work together, allows them to remain focused on the task at hand, and saves them the overhead of learning whole new systems.“ [Hupfer et al. 2004, s.4]

3.2.1.1 Vorteile

Wird es den Anwendern ermöglicht, die kollaborativen Funktionalitäten direkt aus dem Kontext der operativen Geschäftsanwendungen zu starten, so bietet das folgende Vorteile für den Anwender:

- Der Anwender erfährt **keine Medienbrüche**, da der Übergang vom Ort der Problementstehung zum Kollaborationswerkzeug fließend ist. [vgl. Huang 2004, S.105]. Mahowald stellt diesen fließenden Übergang („seamless to users“) als Benutzerfreundlichkeit heraus. [vgl. Mahowald 2001, S.2]
- Der Anwender **behält den Fokus** auf die gerade zu erledigende Aufgabe. Wenn ein Anwender seinen Arbeitskontext verlassen muss, um ein Kollaborationswerkzeug zu benutzen, so erfährt er nicht nur Medienbrüche, sondern Inhalte in diesen Kollaborationswerkzeugen können seinen Fokus ganz auf andere Sachverhalte ablenken. [vgl. Mahowald 2001, S.2]
- Die Kollaboration aus dem Arbeitskontext kommt der **natürlichen Kollaborationsweise** näher. Der Anwender muss sein Kollaborationsverhalten nicht den Gegebenheiten seiner Kollaborationswerkzeuge anpassen. Boyd bezeichnet die Kollaboration außerhalb des Kontextes als unnatürlich. „My contention is that the unnaturalsness of out-of-context collaboration [...] is the direct consequence of the way the applications channel human communication.“ [Boyd 2003, S.1]

3.2.1.2 Anforderungen

Die erste Stufe des Modells, *Starten kollaborativer Funktionen aus dem Arbeitskontext*, stellt daher die folgenden Anforderungen an ein kontextuell kollaboratives System:

- Bereitstellung **vielfältiger kollaborative Funktionen** zur synchronen und asynchronen Kollaboration wie z.B. E-Mail, Instant Messaging, Chat, Diskussionsforen, E-Meeting, gemeinsam genutzte Dokumentendatenbanken etc.
- Bereitstellung dieser kollaborativen Funktionalität als individuelle Services bzw. funktionale Einheiten. **Zerlegen** von monolithischen Kollaborationssystemen und Applikationen **in separate Einheiten**.
- Integration dieser kollaborativen Komponenten **in bestehende Business-Applikationen** (vgl. z.B. Use-Cases 3.1.3, 3.1.10 oder 3.1.13)
- Möglichkeit der Zusammenstellung kollaborativer Komponenten und Business Komponenten zu **neuen, aggregierten Business-Applikationen** (vgl. Use-Case 3.1.9)

3.2.1.3 Systemmerkmale

Um diese Anforderungen zu erfüllen sollte ein idealtypisches Framework für kontextuelle Kollaboration die folgenden Systemmerkmale bereitstellen:

- Ein standardbasiertes **Komponentenmodell**, auf dessen Basis die kollaborativen Funktionalitäten implementiert und bereitgestellt werden können,
- Verschiedene Mechanismen zur **Komponentenintegration**, die es erlauben die kollaborativen Komponenten in bestehende oder neue Business-Applikationen zu integrieren und daraus aufzurufen, z.B.
 - die Integration auf API-Ebene
 - die Integration auf Businessobjektebene - z.B. in das Personenobjekt (vgl. Use-Cases 3.1.3, 3.1.10, 3.1.12, 3.1.13) oder in ein Dokument (vgl. Use-Case 3.1.13)
 - die Integration auf Komponentenebene (vgl. Use-Cases 3.1.5, 3.1.9)
- Ein Mechanismus für **Single-Sign-On (SSO)**. Bei einem System, welches aus vielen verschiedenen Komponenten mit jeweils verschiedenen Authentifizierungsmechanismen zusammengesetzt ist, ermöglicht SSO dem Anwender sich einmal am System anzumelden und dann automatisch an allen Komponenten authentifiziert zu werden, ohne sich mehrmals anmelden zu müssen (vgl. Use-Case 3.1.10)

3.2.2 Stufe 2: Einbinden des Arbeitskontextes in die Kollaboration

Ein weiterer Nachteil der Kollaboration außerhalb des Arbeitskontextes liegt darin, dass der Anwender zum Zwecke der Kollaboration die relevanten Kontextinformationen erneut rekonstruieren muss. Hat der Anwender z.B. eine Rückfrage an die Person, die eine Bestellung aufgenommen hat, die der Anwender gerade bearbeitet, so muss er nicht nur seinen Mail-Client starten, sondern zusätzlich die Mail-Adresse des Kollegen herausfinden und in die Mail eintragen. Befindet sich der Kollege nicht zufällig in seinem persönlichen Adressbuch so muss der Anwender evtl. das unternehmensweite Adressbuch durchsuchen, um die entsprechende Adresse zu finden. Zudem kann es in großen Unternehmen sein, dass mehrere Kollegen den gleichen oder einen ähnlichen Namen haben, was durchaus zu Fehlsendungen der Mail führen kann. Will sich der Anwender

in seiner Mail auf die Bestellung beziehen, so muss er zunächst manuell die Referenz auf die entsprechende Bestellung, z.B. anhand der Bestellnummer oder der Kundennummer, in die Mail eingeben. Ebenso muss sein Kollege dann erst die entsprechende Bestellung aus dem System suchen. All dies ist aufwendig und fehleranfällig.

Dieses Problem kann die kontextuelle Kollaboration in der zweiten Stufe adressieren. Durch die Integration der kollaborativen Funktionalität in den Kontext der Business-Applikation besteht die Möglichkeit die relevanten Kontextinformationen automatisch mit in die Kollaborationsfunktionen zu übernehmen. „Once collaboration has been integrated into the context of core application in unobtrusive ways, the next level of integration can be considered-insinuating context into the collaborative features.“ [Hupfer et al. 2004, S.4] So kann z.B. beim Starten einer Mail aus dem Kontext der Bestellung automatisch der Autor der Bestellung als Adressat in der neuen Mail eingetragen werden. Der Anwender muss nicht erst in seinem Adressbuch nach ihm suchen. Ebenso kann z.B. der Inhalt eines Dokumentes, bzw. ein Link auf das Dokument direkt beim Start eines Chats mit übergeben werden, so dass der Kommunikationspartner durch einen Klick auf den Link direkt zu dem entsprechenden Dokument gelangt. „Zugleich können die Informationen aus dem Aufgabenerfüllungskontext bei der Kommunikation bzw. Kollaboration Betrachtung finden, welches für eine bessere Hilfestellung und Orientierung bei der Kommunikation bzw. Kollaboration sorgt.“ [Huang 2004, S.106] Für die Anwender ergibt sich dadurch ein durchgängiger Arbeitsfluss, da die Arbeit inklusive der Kollaboration vollständig innerhalb des Prozesskontextes erledigt werden kann. Dies wirkt sich Effizienz steigernd auf die Bearbeitung des entsprechenden Geschäftsprozesses aus.

3.2.2.1 Vorteile

Durch das automatische Einbinden von Arbeitskontext in die Kollaboration ergeben sich folgende Vorteile für die die Kollaboration. Die Vorteile dieser Stufe der kontextuellen Kollaboration sind:

- **Kein Kontextverlust**, d.h. der Kontext zur Kollaboration muss nicht von den Kollaborationspartnern manuell hergestellt werden.
- **Bessere Orientierung** des Anwenders. Alle relevanten Kontextinformationen können automatisch übernommen werden.

- **Geringe Hürde** für den Anwender, **einen Kollaborationsvorgang zu starten**, da er keinen oder wenig Aufwand betreiben muss, den Kontext für seine Kollaborationspartner herzustellen.
- **Effizienterer Arbeitsfluss**, da die Kollaboration integraler Bestandteil der Geschäftsprozessbearbeitung wird und nicht mehr ein separater Vorgang ist.

3.2.2.2 Anforderungen

Die zweite Stufe des Modells, *Einbinden des Arbeitskontextes in die Kollaboration*, stellt daher die folgenden, zusätzlichen Anforderungen an das System:

- **Übergabe von Kontextinformationen** von der Business-Applikation an die aufgerufene kollaborative Komponente (vgl. Use-Case 3.1.3)
- **Austausch von Kontextinformationen** zwischen zwei Komponenten (vgl. Use-Case 3.1.5)

3.2.2.3 Systemmerkmale

Um diese Stufe der kontextuellen Kollaboration zu realisieren, sollte ein idealtypisches Framework folgende Systemmerkmale aufweisen:

- Standardisierte Mechanismen zur **Interkomponentenkommunikation**, die es verschiedenen Komponenten ermöglicht Kontextinformationen untereinander oder mit der Business-Applikation auszutauschen um sich zu kontextualisieren.

3.2.3 Stufe 3: Persistierung der Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen

Wird die Kollaboration aus den separaten Kollaborationsapplikationen und nicht aus dem Kontext der Business-Applikation oder des Geschäftsprozesses heraus gestartet, so können die dabei entstehenden Kollaborationsergebnisse und Informationen typischerweise auch nicht mit dem Kontext assoziiert oder in ihm gespeichert werden. Informationen zu einem Geschäftsprozess, die z.B. in E-Mails ausgetauscht werden, werden dann zusammen mit anderen E-Mails in einer Mail-Datei oder einer Mail-Datenbank einzelner Personen gespeichert. Das Kontextwissen, dass diese Information zu dem speziellen Geschäftsprozess gehört, existiert dann nur implizit und im Kopf der Personen, die an der Kommunikation – also dem Mailverkehr – beteiligt waren. Zusätzlich sind diese

3. Anforderungen und Konzepte

Informationen für andere, nicht an dem Mailverkehr beteiligten Personen, nicht zugreifbar. Jegliche zusätzliche kontextuelle Information wie zum Beispiel eine Nachfrage zur Klärung eines Sachverhaltes oder die Argumentation, die zu einer bestimmten Entscheidung führte, sowie auch ein unterstützendes Dokument mit Hintergrund- oder Referenzinformationen, das für den Geschäftsprozess relevant wäre, sind somit oft nicht mehr zugreifbar. Damit entstehen Informations- und Wissensinseln die nicht mit dem entsprechenden Prozess in Beziehung gebracht werden können und nicht von Dritten einsehbar oder auswertbar sind. So werden alle Mails in Maildateien gespeichert, alle Dokumente in Dokumentenbibliotheken und Geschäftsobjekte wie z.B. Rechnungen, in Rechnungsdatenbanken oder den entsprechenden Business-Applikationen, mit denen sie erstellt wurden. Aus Sicht des Anwenders in dem entsprechenden Geschäftsprozess führt dies zu einer starken Verstreuung der zu dem Prozess gehörenden Informationen. Da diese Informationen nicht im Kontext des jeweiligen Prozesses erzeugt wurden, kann auch keine explizite Referenz zwischen den Informationsobjekten und dem Geschäftsprozess hergestellt oder gespeichert werden. Daher gibt es auch keine einfache Möglichkeit, eine prozessorientierte Sicht auf die Informationen zu bekommen, d.h. es ist nicht möglich zu bestimmen, welche Dokumente, Informationen oder andere Ressourcen zu einem bestimmten Geschäftsprozess gehören oder mit ihm in Beziehung stehen.

Die dritte Stufe der kontextuellen Kollaboration beschreibt daher die Persistierung der Beziehungen, die zwischen Kollaborationsergebnissen, z.B. einer E-Mail, und dem Geschäftsprozess bisher nicht explizit existierte, im Kontext des Geschäftsprozesses selbst. Die Möglichkeit, alle relevanten Informationen einen Geschäftsprozess betreffend auch im Kontext des Geschäftsprozesses zu bewahren, bietet dem Anwender eine Nachhaltigkeit der kollaborativen Vorgänge und Entscheidungen. Dabei können drei unterschiedliche Szenarien unterschieden werden:

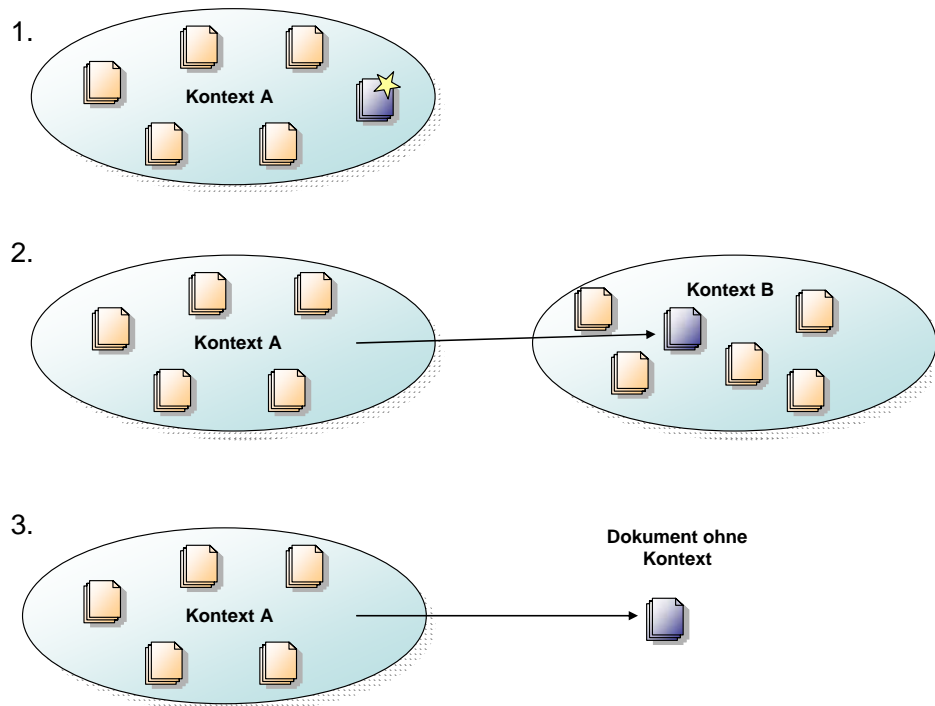


Abb. 7: Drei Arten von Kontextbeziehungen

1. Persistierung von Kontextbeziehungen zu Kollaborationsdokumenten, die im Kontext, d.h. im Verlauf der Bearbeitung des Geschäftsprozesses entstehen.
2. Herstellen von Kontextbeziehung zu Dokumenten und Objekten, die zum Kontext eines anderen Geschäftsprozesses gehören und zusätzlich einem anderen Geschäftsprozess angehören sollen.
3. Hinzufügen von Kontextbeziehungen zu Dokumenten und Objekten, die bisher keinem Geschäftsprozess angehören.

Der oben verwendete Begriff „Objekt“ bezeichnet hier alle möglichen referenzierbaren Entitäten eines Unternehmens und schließt auch Personen mit ein. Dokumente, Prozesse, E-Mails, oder auch Shared-Spaces sind weitere Beispiele für Objekte.

Es wird deutlich, dass die Anforderung, Kollaborationsfunktionalitäten in operative Applikationen zu integrieren, nur ein erster Schritt für die kontextuelle Kollaboration sein kann.

„But consider what happens when one later wants to retrieve the discussion. Was it an email or a chat? Who saved it and where? Even if the transcripts and emails can be located and rummaged through, the work they reference may not be obvious, because the discussions are completely decoupled from the work artifacts.”[Hupfer et al. 2004,

S.2] Durch das explizite Erstellen von Beziehungen zwischen Kollaborationsergebnissen und dem eigentlichen Geschäftsprozess kann dieses implizite Wissen externalisiert werden. [vgl. Wegener 2002, S.15ff]

Zur Persistierung verschiedenster Objekte und Objektbeziehungen im Kontext eines Geschäftsprozesses eignen sich vor allem gemeinsam genutzte Plätze (Shared-Spaces). Sie basieren meist auf einer gemeinsam genutzten Datenbank oder Dokumentenbibliothek. Damit wird der Shared-Space zu einem Container, der den Kontext des Geschäftsprozesses repräsentiert.

Ist die Möglichkeit geschaffen diese Kontextbeziehungen herzustellen und zu persistieren, so können nicht nur die Objekte referenziert werden, die direkt als Teil des Geschäftsprozesses erstellt wurden, sondern auch die Objekte, die außerhalb dieses Kontextes entstanden sind bzw. einem anderen Kontext angehören. Ein Beispiel hierfür ist ein Dokument, das im Rahmen der Geschäftsprozessabwicklung entstanden ist und nun relevant für einen anderen Prozess wird. Diese neue Beziehung kann in dem neuen Prozess zusätzlich referenziert werden.

Dieses grundsätzliche Prinzip der Beziehungserstellung durch Referenzierung lässt sich dann auch auf Dokumente und Informationsobjekte anwenden, die nicht im Rahmen eines bestimmten Geschäftsprozesses entstanden sind, z.B. ein Dokument mit betrieblichen Richtlinien, die den Prozess betreffen. Diese Dokumente und Objekte lassen sich so mit einem bestehenden Geschäftsprozess assoziieren und als Referenz in diesem Kontext speichern.

3.2.3.1 Vorteile

Durch die Persistierung der Kontextbeziehungen auf einen Geschäftsprozess ergibt sich folglich eine Reihe von Vorteilen:

- Durch die Referenzierung aller Kontextbeziehungen auf den Geschäftsprozess ist es dem Anwender möglich, alle Informationen, die benötigt werden, um den Geschäftsprozess zu bearbeiten oder die in seinem Verlauf entstanden sind, von einem Punkt aus einzusehen und zu erreichen (**Single Point of Access**).
- Durch den direkten Zugriff auf alle mit dem Geschäftsprozess verbundenen Informationen können **fundierte Entscheidungen** getroffen werden.

- Ebenso werden getroffene Entscheidungen nachträglich **besser nachvollziehbar**.
- Auf Anfragen bezüglich eines Geschäftsprozesses, z.B. vom Kunden, kann **schneller und besser informiert** reagiert werden.
- Für den Anwender ergibt sich eine **Transparenz und Übersicht** über den gesamten Arbeitsprozess
- Diese Übersicht führt zu einer **prozessorientierten Arbeitsweise**. Dabei sollte nicht die Individualleistung der einzelnen Anwender im Vordergrund stehen, sondern das Prozessergebnis. „Collaboration is viewed as successful based on the process outcome, not the output of individuals within various tasks.” [vgl. Gotta 2004, S. 2]

3.2.3.2 Anforderungen

Aus dieser dritten Stufe des Modells, *Persistierung der Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen*, ergeben sich folgende Anforderungen an ein kontextuell-kollaboratives System:

- Möglichkeit als Anwender einen **neuen, gemeinsam genutzten Kontext** für die Bearbeitung eines Geschäftsprozesses zu erstellen (vgl. Use-Case 3.1.4, 3.1.8)
- Möglichkeit, die **Mitglieder** eines gemeinsam genutzten Kontextes zu definieren, entweder implizit durch das Starten einer Kollaboration (vgl. Use-Case 3.1.4) oder explizit durch Hinzufügen von Personen zu einem bestehenden Kontext (vgl. Use-Case 3.1.9)
- Neue Dokumente zur **gemeinsamen asynchronen Nutzung** in einem bestehenden Kontext anlegen (vgl. Use-Case 3.1.4) oder zu einem bestehenden Kontext hinzufügen (vgl. Use-Cases 3.1.6, 3.1.7)
- Gemeinsames **synchrones Bearbeiten** von Dokumenten und Ressourcen aus dem Arbeitskontext durch die Mitglieder (Vgl. Use-Case 3.1.13)
- **Sichtbarkeit des Bearbeitungszustandes aller Dokumente** bzw. der **Verfügbarkeit von Personen**, die mit dem Arbeitskontext in Beziehung stehen (vgl. Use-Cases 3.1.10, 3.1.12)

- **Benachrichtigung und Abonnie- rung** für wichtige Ereignisse und Statusänderungen an Personen, Ressourcen und Kontexten (vgl. Use-Case 3.1.7, 3.1.9)

3.2.3.3 Systemmerkmale

Um diese Anforderungen zu erfüllen sollte ein idealtypisches Framework für kontextuelle Kollaboration die folgenden Systemmerkmale bereitstellen:

- Verschiedene Arten von **Shared-Spaces** als persistente Kontexte für unterschiedliche Geschäftsprozessstypen z.B.:
 - Ad-hoc orientierte, Activity-zentrierte-Spaces
 - Semistrukturierte Shared-Workspaces
 - Formale, Workflow getriebene Spaces
- **Asynchrones Sharing** von Objekten im Kontext eines Geschäftsprozesses.
- **Synchrones Sharing**, um Objekte im Kontext eines Geschäftsprozesses in Echtzeit gemeinsam zu nutzen oder zu bearbeiten.
- Eine Liste von **Mitgliedern** für jeden Shared-Space, welche die Personen aufnimmt, die an dem Geschäftsprozess beteiligt sind und die in diesem Kontext bestehenden Dokumente gemeinsam nutzen können.
- **Kontext-Awareness** als Realtime-Mechanismus um über den Online-Status anderer Mitglieder oder den Bearbeitungsstatus eines Dokumentes im Kontext des Prozesses informiert zu sein.
- **Notification** als Mechanismus zur Benachrichtigung über Statusänderungen wenn der Anwender sich gerade nicht im Kontext des Prozesses befindet.
- Ein einheitliches Datenschema als Teil des **Metadatenmodells** zur Repräsentation von Objekt und Referenzen, so dass Kontextobjekte und Kontextbeziehungen einheitlich referenziert, gespeichert und ausgelesen werden können.

3.2.4 Stufe 4: Navigation durch das multidimensionale Geflecht von Kontextbeziehungen

Die oben beschriebene Persistierung der Beziehungen erfolgt bei der kontextuellen Kollaboration in der Regel im Kontext von Geschäftsprozessen, in denen sie entstanden sind. Während dies, wie oben beschrieben, viele Vorteile bringt, kann es allerdings auch

3. Anforderungen und Konzepte

wieder zur Bildung von Informations- und Wissensinseln rund um Geschäftsprozesse führen. So kann ein Anwender zwar einfach sehen, welche Dokumente und Informationsobjekte, wie etwa ein Angebot, zu einem bestimmten Prozess gehören, aber es ist z.B. wiederum nicht einfach zu sehen, welche anderen Angebote bisher für den gleichen Kunden erstellt wurden.

Es kann aber wichtig sein, den jeweiligen Kontext von Geschäftsprozessen zu verlassen und einen Blick auf die entstandenen Beziehungen in einem anderen Kontext zu werfen. So kann es z.B. für einen Manager hilfreich sein, die Beziehungen aus dem Kontext von Personen zu betrachten. D.h. er kann dann für eine bestimmte Person nachvollziehen, mit welchen Prozessen sie auf welche Art in Beziehung steht. Oder er kann sich alle Prozesse anzeigen lassen, die gerade in seiner eigenen Verantwortung liegen und noch nicht abgeschlossen sind. Dann betrachtet er Prozesse aus dem Kontext seiner eigenen Person. Das bedeutet, Beziehungen entstehen im Unternehmen nicht nur von einem Geschäftsprozess zu anderen Businessobjekten, sondern auch zwischen diesen Objekten, z.B. zwischen Personen oder Dokumenten untereinander.

Werden auch diese Beziehungen persistiert, so entsteht bei der kollaborativen Arbeit ein multidimensionales Beziehungsgeflecht zwischen den verschiedenen Geschäftsobjekten des Unternehmens. Auch wenn die Beziehungen in der Regel im Kontext von Geschäftsprozessen entstehen, so sollen diese Beziehungen aus einem anderen Kontext betrachtet werden können. „Kontext“ bezeichnet hier das jeweilige Geschäftsobjekt, welches aktuell im Fokus des Betrachters liegt. Die grundsätzlichen Bereiche, denen ein Geschäftsobjekt zugehörig ist, ergeben sich dabei aus der Organisationsstruktur von Unternehmen. [Vergleiche hierzu Kapitel 2.2]

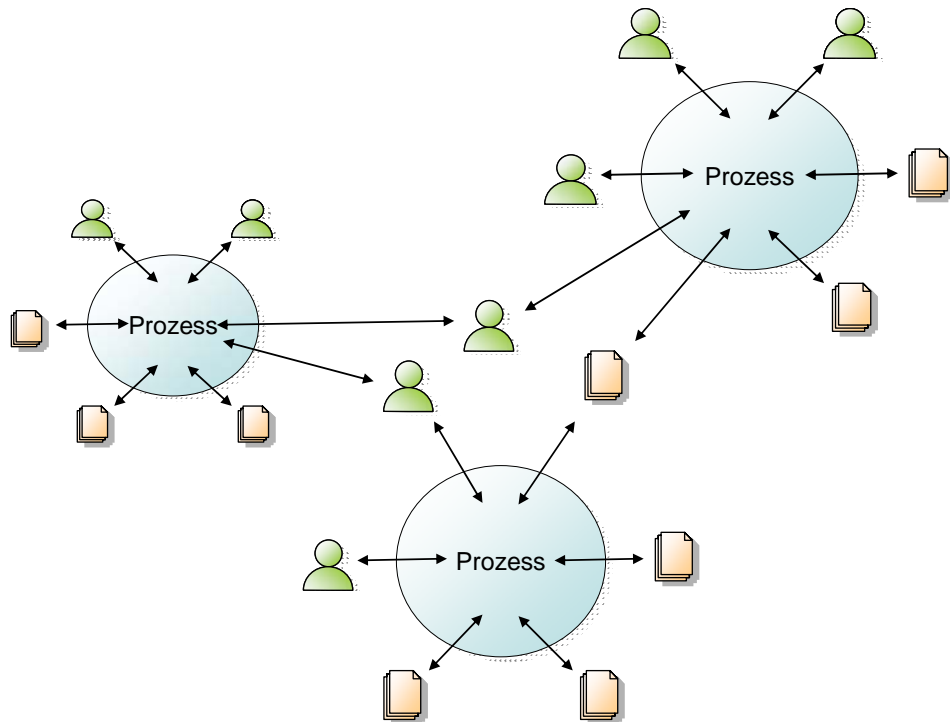


Abb. 8: Multidimensionales Geflecht von Kontextbeziehungen

Dieses Beziehungsgeflecht beschreibt damit einen Zusammenhang zwischen Prozessen, an denen gearbeitet wird, den Personen, die daran beteiligt sind, den Applikationen, die die Personen benutzen, um diese Prozesse voranzutreiben sowie den Informationen, die für die Abwicklung von den Geschäftsprozessen nötig sind. Dabei beschreibt es auch Zusammenhänge, wie z.B. welche Mitarbeiter miteinander kollaborieren, oder dass zwei Dokumente miteinander in Beziehung stehen, weil sie die gleiche Person als Autor haben.

Die Multidimensionalität dieses Beziehungsgeflechts ergibt sich daraus, dass nicht nur Beziehungen zwischen Elementen verschiedener Dimensionen, d.h. zwischen Objekten verschiedener Typen gibt, sondern die Beziehungen bestehen auch innerhalb der Dimensionen, z.B. sind Personen auch mit anderen Personen verbunden. Viele Dokumente können mit einer Person verbunden sein, z.B. weil sie der Autor dieser Dokumente ist. Dadurch stehen die Dokumente auch untereinander in Beziehung.

Werden all diese Beziehungen im Kontext der Geschäftsprozesse gespeichert und damit externalisiert, so lassen sich diese Beziehungen auswerten und vom Anwender navigieren. Er kann so das multidimensionale Geflecht von Kontextbeziehungen Schritt für Schritt entlang von Beziehungspfaden navigieren und so wichtige Informationen über die kontextuellen Zusammenhänge um seinem Arbeitskontext erhalten.

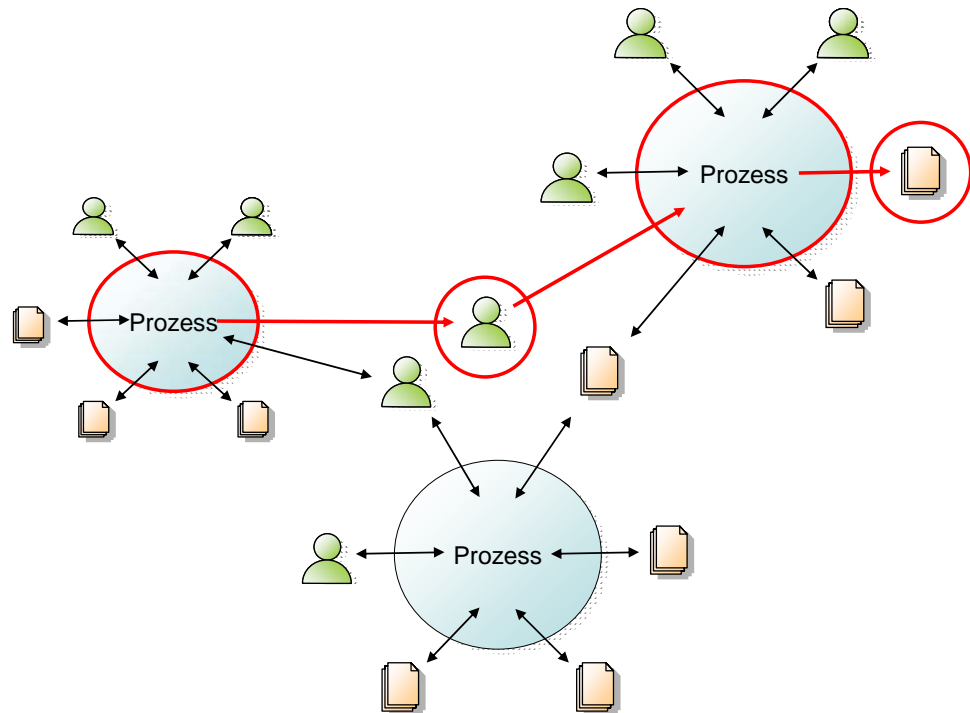


Abb. 9: Navigationspfad über Kontextbeziehungen

3.2.4.1 Vorteile

Die Vorteile der Kontextnavigation lassen sich folgendermaßen zusammenfassen:

- **Transparenz von Zusammenhängen** über einen speziellen Geschäftsprozess hinaus.
- Möglichkeit der **Analyse von kontextuellen Zusammenhängen**. Das bedeutet, Mitarbeiter sind dadurch in der Lage kontextuelle Zusammenhänge und kontextuelle Nähe von Objekten innerhalb eines Unternehmens zu erkennen.

3.2.4.2 Anforderungen

Die vierte Stufe des Modells zur kontextuellen Kollaboration stellt schließlich die folgenden Anforderungen an das System:

- Möglichkeit zur **Navigation** zwischen verschiedenen Kontexten über gemeinsam referenzierte Personen, Informationen oder Prozesse (vgl. Use-Case 3.1.11)
- Möglichkeit zur **Sichtenbildung** über verschiedene Aspekte der Kontexte und Kontextbeziehungen

3.2.4.3 Systemmerkmale

Die folgenden Systemmerkmale stellen die Voraussetzung für die Kontextnavigation und Sichtenbildung dar:

- Einheitliche Adressierbarkeit und Navigierbarkeit von Objekten und Ressourcen durch ein standardisiertes **Metadatenmodell** sowie eine konsistente Speicherung von Objekten und Objektbeziehungen als Teil des Kontextes.
- Mechanismen zur **Kontextnavigation**

3.3 Strukturmerkmale eines idealtypischen Frameworks für kontextuelle Kollaboration

Im vorigen Kapitel wurden die Konzepte der kontextuellen Kollaboration und das entwickelte „Vier-Stufenmodell der kontextuellen Kollaboration“ vorgestellt. Die daraus identifizierten Anforderungen an ein kontextuell-kollaboratives Framework sind als Strukturmerkmale eines idealtypischen Frameworks benannt worden. In diesem Kapitel werden die Eigenschaften und die Bedeutung dieser Strukturmerkmale für ein solches Framework herausgearbeitet und erläutert.

Abb. 10 beschreibt verschiedene Schichten eines idealtypischen Frameworks für die kontextuelle Kollaboration, so wie diese im Rahmen der Arbeit verstanden werden. Das Framework dient als Grundlage zur Einordnung der nachfolgend beschriebenen Strukturmerkmale.

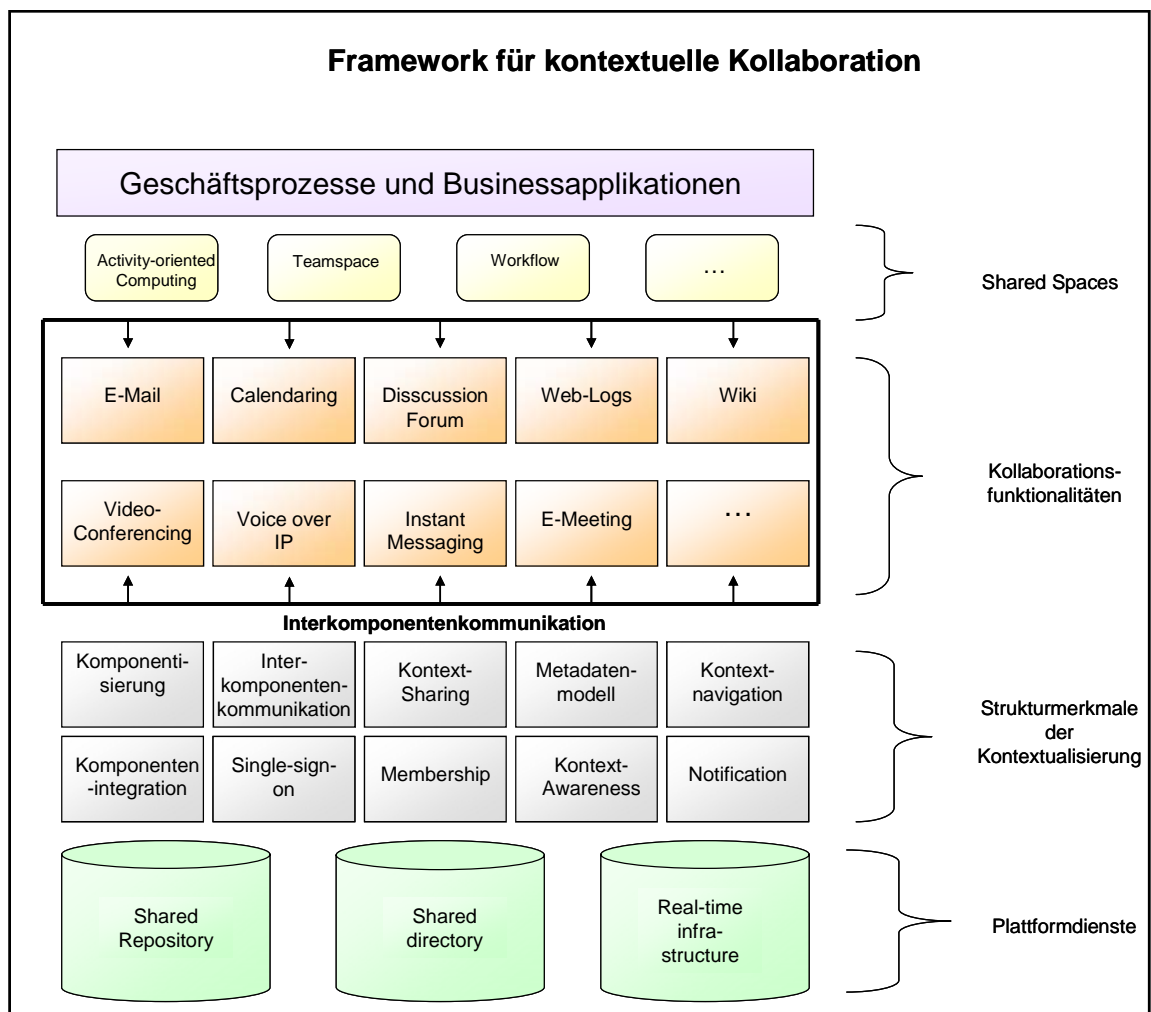


Abb. 10: Architektur eines idealtypischen Frameworks zur kontextuellen Kollaboration

3. Anforderungen und Konzepte

Als Basis des Framework der kontextuellen Kollaboration gelten **die Plattformdienste** einer kollaborativen Architektur: *Shared Repository*, *Shared Directory* und die *Real-time Infrastructure*. Nastansky sieht in einer Architektur mit verteilten Datenbanken ein Kernelement für kollaborative Systeme. Die gemeinsame Nutzung von Daten ist dabei Grundvoraussetzung für eine flexible Kollaborationsarchitektur. Maßgeblich unterstützt werden diese Datenbanken durch intelligente Replikationsmechanismen um Datenkonsistenzen zu gewahren. [vgl. Nastansky et al. 2000, S.250]

Im Rahmen dieser Arbeit werden die Strukturmerkmale abstrahiert von diesen Plattformdiensten betrachtet. Es wird daher die Annahme getroffen, dass ihre, für die Kollaboration wichtigen, Funktionen genutzt werden können. Folglich wird nur eine kurze Erläuterung ihrer Funktionen gegeben.

- Das **Shared Repository** dient der Persistierung gemeinsam genutzter Daten und Informationen und bildet die Basis für asynchrone Share-Funktionalitäten.
- Die **Realtime-Infrastruktur** bildet die Basis für synchrone Share-Funktionalitäten sowie für die Awareness-Funktionalitäten
- Das **Shared Directory** dient dazu, die Benutzer gegenüber dem System zu authentifizieren als auch deren Metadaten wie z.B. eine E-Mail-Adresse und Instant Messaging-Daten zu speichern und für die kollaborativen Funktionalitäten zur Verfügung zu stellen.

Die zweite dargestellte Schicht des Frameworks bilden die **Strukturmerkmale der Kontextualisierung**. Sie geben einem kontextuellen Kollaborationssystem die nötigen Strukturen zur Kontextualisierung. Durch die Funktionsweisen der Strukturmerkmale hebt sich das kontextuelle Kollaborationssystem von herkömmlichen Systemen ab. Ihre Spezifikation bildet den Hauptteil des Kapitels 3.3.

Die in der dritten Schicht angesiedelten **kollaborativen Funktionalitäten**, wie E-Mail oder Instant Messaging, bilden die Grundlage der Kollaborationsvorgänge in einem kontextuell-kollaborativen System. Ihre Bedeutung wird in Kapitel 3.3.1 erläutert.

Die vierte Schicht wird gebildet von aggregierten Applikationen zur gemeinsamen Bearbeitung von Geschäftsprozessen (*Shared Spaces*). Ihre Bedeutung für die kontextuelle Kollaboration wird in Kapitel 3.3.5.2 beschrieben.

In der obersten Schicht sind im Sinne der kontextuellen Kollaboration idealtypisch die *Business-Applikationen* angesiedelt. Diese können dann die in tieferen Schichten lie-

genden kollaborativen Funktionalitäten und Strukturmerkmale zur kontextuellen Kollaboration nutzen. Beispiele hierfür sind CRM-Systeme, ERP-Systeme oder auch SCM-Systeme. Die Untersuchung der Business-Applikationsschicht ist nicht Gegenstand dieser Arbeit.

3.3.1 Kollaborative Funktionalitäten

Da die kontextuelle Kollaboration von ihren Wesen her ein besonderes Konzept der Kollaboration darstellt, sind kollaborative Funktionalitäten die Grundlage der kontextuellen Kollaboration. Der Begriff kollaborative Funktionalität schließt definitionsgemäß die Kommunikation mit ein. Es sei erwähnt, dass einige der nachfolgend genannten Funktionalitäten in den meisten Fällen mit Applikationen assoziiert werden, die typischerweise wieder eine Reihe von Funktionalitäten beinhalten. Als Beispiel sei hier Instant Messaging genannt. Viele der am Markt befindlichen Instant Messaging-Tools haben auch Awareness- und Screen-Sharing Funktionalitäten realisiert. Es wird hier aber jeweils die Kernkollaborationsfunktionalität angesprochen, was nochmals unterstreichen soll, dass die Kollaborationsfunktionalitäten keine monolithischen Applikationen darstellen, sondern integrierbare Funktionen.

Zu diesen kollaborativen Funktionalitäten gehören sowohl synchrone als auch asynchrone Kollaborationsfunktionen. Die synchronen Kommunikations- und Kollaborationsfunktionalitäten, auch als Realtime-Funktionalitäten bezeichnet, erhalten in der kontextuellen Kollaboration einen besonderen Stellenwert. Einerseits geben sie einem kontextuell-kollaborativen System die Agilität und Flexibilität schnell auf Veränderungen und Probleme reagieren zu können, andererseits findet ihr effektiver Einsatz durch die genannten Strukturmerkmale eines kontextuell-kollaborativen Systems erst volle Entfaltung.

Beispiele für **synchrone kollaborative Funktionen** sind:

- **Instant Messaging:** „Mittels Instant Messaging, adäquat einem Echtzeit-Chat, können *E-Mails* und *Nachrichten*, aber auch Bilder, Audio- und Video-Files ausgetauscht werden. Der Nachrichtenaustausch ist unmittelbar und verkürzt die Kommunikationsprozesse.“ [ITWissen]

- **Videokonferenz** (video teleconferencing): „Unter Videokonferenz-Systemen werden solche Systeme verstanden, bei denen eine Teilnehmergruppe mit einer zweiten, entfernten Teilnehmergruppe in *Echtzeit* visuell und sprachlich kommuniziert. In der Regel können bei einer Videokonferenz neben der *Kommunikation* über den Blickkontakt der Teilnehmer auch *Dokumente* oder Bewegbilddarstellungen übertragen werden. Diese Möglichkeiten unterscheiden Videokonferenzen von der *Bildtelefonie*.“ [ITWissen]
- **VoIP**: „Abk. Voice over IP. Auch mit "Internet-Telefonie" und "Voice over Net (VoN)" bezeichneter anwendungsnaher Telekommunikationsdienst zur Übertragung von Sprache über IP-basierte Netze (Internet Protocol), wie das weltweite Internet oder unternehmensinterne Netze gleicher Technologie (Intranets).“ [IT-Fachbegriffe]

Asynchrone Kollaborationswerkzeuge sind z.B.:

- **E-Mail**: „Bei E-Mail, der elektronischen Post, handelt es sich um ein personenbezogenes *Kommunikationssystem* für den Versand von elektronischen Mitteilungen in *Form* von Briefen, Memos und anderen *Texten*, an die ein *Anhang* angehängt werden kann.“ [ITWissen]
- **Diskussionsforum**: (newsgroup) „Diskussionsforen sind themenbezogene öffentliche Diskussionsrunden, die nicht in *Echtzeit* ablaufen, sondern zeitversetzt. Im WWW und Usenet spricht man im Allgemeinen von Newsgroups, von denen es tausende gibt, zu manchen Themen mehrere hundert.“
- **Blog**: (Weblog) „Blog ist aus der Bezeichnung Weblog abgeleitet. Das Weblog wiederum ist eine Art Logbuch für das Web. Dieses Logbuch wird wie ein Journal geführt und mit Einträgen, Kommentaren und Notizen versehen, die chronologisch geordnet sind: der aktuellste Eintrag befindet sich an oberster Stelle. Für die Erstellung und Pflege des Blogs steht entsprechende Blog-Software zur Verfügung. Die Weblog-Einträge können von allen Lesern kommentiert werden.“ [ITWissen]

Die unterschiedlichen Formen und Ausprägungen kollaborativer Funktionalitäten sind nicht Gegenstand der Untersuchung im Rahmen dieser Arbeit. Es wird als Voraussetzung angesehen, dass die Funktionalitäten grundsätzlich bereit stehen können, bzw. implementierbar sind für ein idealtypisches Framework der kontextuellen Kollaborati-

on. Die hier genannten Funktionalitäten sind dabei nur Beispiele. Grundsätzlich wird im Rahmen dieser Arbeit jede elektronisch unterstützte Kollaborations- und Kommunikationsfunktionalität als für die kontextuelle Kollaboration geeignet angesehen.

3.3.2 Komponentisierung

Um ein wesentliches Merkmal der kontextuellen Kollaboration zu erfüllen, nämlich die Integration kollaborativer Funktionalitäten in die Business-Applikationen, ist es nötig, diese monolithischen Kollaborationssysteme in kleine, wieder verwendbare Einheiten zu zerlegen, die einzelne, wohl definierte kollaborative Funktionen zur Verfügung stellen.

Sparling definiert eine Komponente und ihre Eigenschaften folgendermaßen: „A component is a language neutral, independently implemented package of software services, delivered in an encapsulated and replaceable container, accessed via one or more published interfaces.”[Sparling 2000, S.47]

Die aus der Definition hervorgehenden Merkmale adressieren damit die Anforderung der *Integrierbarkeit von Funktionalitäten in bestehende Applikationen* an kontextuelle Kollaborationssysteme. Daraus lässt sich schließen, dass die in kontextueller Kollaboration verwendeten Komponenten

- wohldefinierte, dokumentierte Interfaces bzw. APIs zur Verfügung stellen müssen, um sie integrierbar zu machen und
- als unabhängige verkapselte funktionale Einheiten implementiert werden. D.h. sie besitzen keine starken Bindungen zu anderen Komponenten, ohne die sie dann nicht nutzbar wären.

Die komponentenbasierte Applikationsentwicklung sieht bei der Erstellung eines Systems unterschiedliche Rollen der beteiligten Personen für unterschiedliche Phasen des Entwicklungsprozesses vor. Die Abb. 11 fasst die unterschiedlichen Phasen und Rollen zusammen.

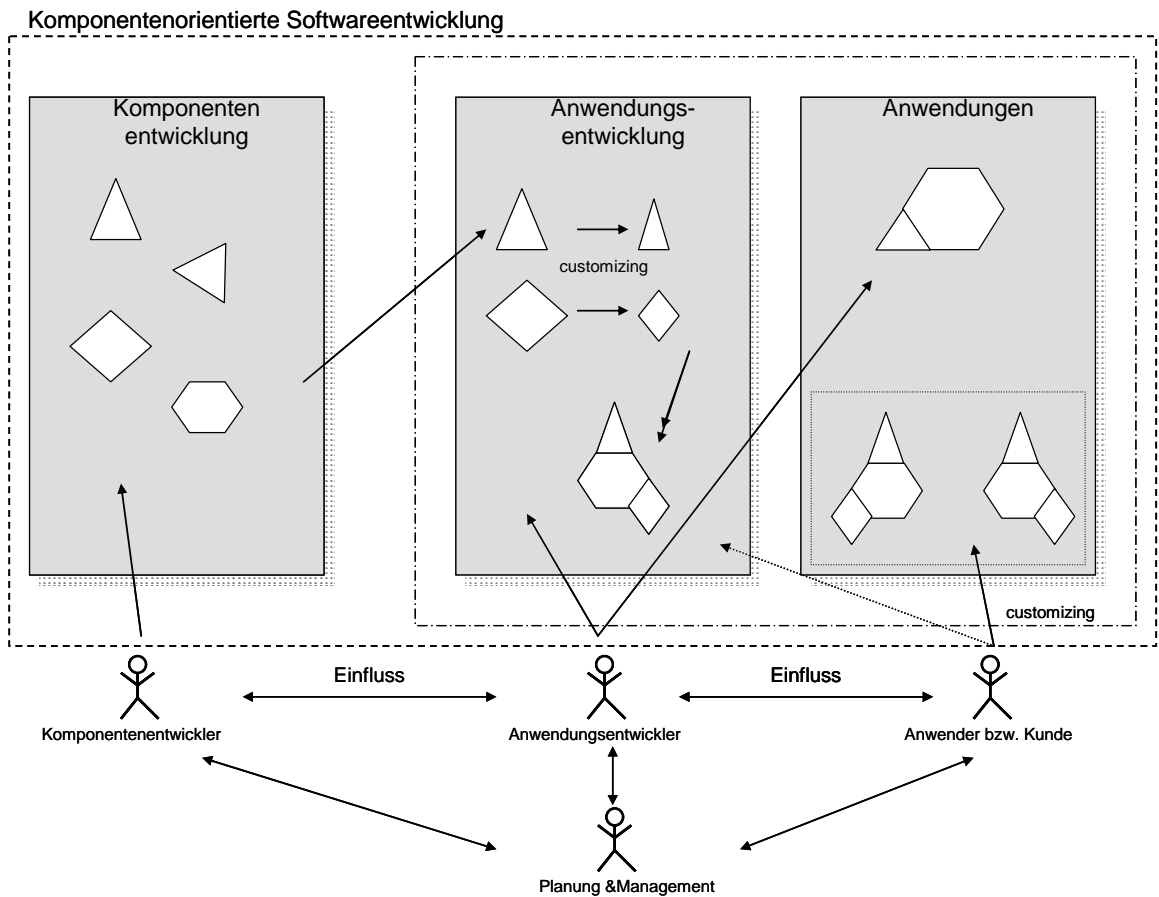


Abb. 11: Der Gesamtprozess einer komponentenbasierten Applikationsentwicklung

Nach [Griffel 1998,S.47]

3.3.3 Komponentenintegration

Die Integration kollaborativer Komponenten ist die Voraussetzung um die Kollaboration aus dem Kontext einer Business-Applikation zu ermöglichen. "A breakthrough in real-time collaboration is the emergence of contextual collaboration - collaboration initiated from within a productivity application or an enterprise portal." [Perey 2003, S.2]

Die Integration von Kollaborations-Komponenten in die Business-Applikation kann auf mehrere Arten geschehen. Im Kontext dieser Arbeit sind insbesondere drei Integrationsarten von Bedeutung:

- Integration auf API-Ebene,
- Integration auf Business-Objektebene
- Integration auf Komponentenebene durch kooperierende Komponenten.

3.3.3.1 Integration auf API-Ebene

Wie in Kapitel 3.3.2 bereits erläutert, bildet die API die Repräsentation einer Komponente nach außen. Dadurch wird z.B. eine Instant-Messaging-Komponente über ihre API von einer Business-Applikation ansprechbar. Die Realisation der Integration durch APIs bietet ein hohes Maß an Flexibilität, da so die Kollaborationsfunktionalitäten in vielfältiger Weise in die Business-Applikation integriert werden kann. Der Applikationsentwickler hat dabei alle typischen Freiheiten diese externe Funktionalität in die Business-Applikation einzubinden. Beispiele hierfür sind Integration einer IM-Komponente in das Userinterface (UI), um dem Anwender an einem bestimmten UI-Punkt Instant-Messaging zur Verfügung zu stellen. Ein anderes Beispiel ist die Integration in den Backend-Code zum Versenden einer Benachrichtigung (engl. Notification) als Instant Message an die Verantwortlichen, wenn der Lagerbestand des Unternehmens eine kritische Marke erreicht hat.

Da bei der Integration auf API-Ebene die Integration auf funktionaler Ebene realisiert wird, zeigt sich die integrierte Funktion oft nicht durch ein eigenes UI (User Interface = Benutzerschnittstelle), sondern sie geht in dem UI der Business-Applikation auf, in die sie integriert wurde. Die Anpassungsmöglichkeiten der Komponente an die Business-Applikation sind daher relativ feingranulär und es ergibt sich für den Anwender ein einheitliches Bild von Business-Applikation und darin integrierter Kollaborationskomponente. Die Integration von Kollaborationskomponenten auf API-Ebene bietet damit einen hohen Grad der Integration.

Die Integration von kollaborativer Funktionalität auf API-Ebene hat aber auch einige Nachteile. So kann die Integration nur durch einen Programmierer erfolgen. Außerdem muss für die Integration der Quellcode der Business-Applikation zur Verfügung stehen in die die kollaborative Funktionalität integriert werden soll. Dies ist insbesondere bei bestehenden Applikationen von Drittherstellern üblicherweise nicht der Fall, so dass sich diese Form der Integration in diesen Fällen nicht anbietet.

Kann die Integration auf API-Ebene durchgeführt werden, z.B. bei selbst erstellten Business-Applikationen, so besteht weiterhin der Nachteil, dass die API der Kollaborationskomponente zum Zeitpunkt der Entwicklung der Business-Applikation bekannt sein muss. Der Entwickler der Business-Applikation muss wissen, welche Kollaborationskomponenten die Business-Applikation unterstützen soll. Das bedeutet aber, dass eine Integration auf API-Ebene von Komponenten, die zum Entwicklungszeitpunkt noch

nicht existieren oder noch nicht bekannt sind, zu einem späteren Zeitpunkt nicht möglich ist. Für Applikationen oder Komponenten, in die eine kollaborative Funktionalität auf die beschriebene Weise integriert wird, wird dadurch eine starke Bindung an die zu integrierende Komponente erzeugt.

Um das Problem der Abhängigkeit zu umgehen, werden in neuerer Zeit für einige Kollaborationskomponenten API-Standards festgelegt. „The current generation of contextual collaboration technology relies primarily on proprietary hooks and APIs. Ratification and adoption of industry standards such as Session Initiation Protocol and SIP for Instant Messaging and Presence Leveraging Extensions will reduce significantly the time involved to integrate real-time collaboration capabilities into third-party applications.”[Perey 2003, S.2] Die Standardisierung vereinheitlicht den Zugriff auf die Funktionalität aller Komponenten, die diese standardisierte API implementieren. Ein solches standardisiertes Interface, welches von verschiedenen Herstellern implementiert werden kann, wird auch Service Provider Interface (SPI) genannt. Durch die Standardisierung ist es möglich, sich von einem speziellen Hersteller unabhängig zu machen. So lässt sich die am besten geeignete Komponente integrieren und auch nachträglich austauschen und durch eine neuere oder passendere Komponente ersetzen. Allerdings müssen die Komponenten gleichen Typs sein. D.h. eine Instant Messaging-Komponente lässt sich nur durch eine andere Instant Messaging-Komponente ersetzen.

Die Standardisierungsgremien bestehen oft aus einem Zusammenschluss einiger großer Softwareunternehmen, z.B. IBM, Microsoft, Bea. Der von ihnen definierte Standard repräsentiert häufig einen kleinsten gemeinsamen Nenner der Funktionalitäten der verschiedenen Hersteller. Daher ist es bei der Integration durch Standard-APIs nicht möglich spezielle Funktionalitäten einer Komponente zu nutzen, die über diesen kleinsten gemeinsamen Nenner hinausgehen.

3.3.3.2 Integration auf Komponentenebene

Eine stärkere Entkopplung von einzelnen Herstellern und die Möglichkeiten der nachträglichen Integration sind die Motivation für eine andere Art der Integration - der Integration auf Komponentenebene. Kernkonzept ist hier die Zusammenstellung von höher aggregierten Komponenten, die neben der verkapselten Funktionalität auch ein eigenständiges User-Interface beinhalten und deshalb manchmal den Charakter „kleiner“ Applikationen haben. Beispiele hierfür sind ein Diskussionsforumskomponente, eine Buddyliste oder eine Dokumentenbibliothek-Komponente. Während diese Komponen-

ten für sich bereits einen Wert haben, liegt das eigentliche Potential darin, dass sie gemeinsam im Kontext einer höherwertigen Applikation z.B. eines Teamspace oder eines Portals integriert werden. Durch die Zusammenstellung in diesen gemeinsamen Kontext entsteht ein Mehrwert der über die Funktionalität der einzelnen Komponente hinausgeht. „Programme werden zu 'Collagen' interagierender Softwarekomponenten (component-based). Ein solches programming in the large [...] bietet den hohen Abstraktionsgrad und die Modellierungsmächtigkeit, die zur Realisierung komplexer Softwaresysteme benötigt wird.“ [Griffel 1998, S.21]. Griffel beschreibt weiter, dass die extremste Ausprägung das sog. Megaprogramming ist. Dabei werden „sehr komplexe und funktional umfangreiche Softwarekomponenten zu übergreifenden Anwendungssystemen kombiniert.“ [vgl. Griffel 1998, S.21]

Die in Kapitel 3.1 vorgestellten Szenarien machen deutlich, dass es für den Einsatzbereich von kontextuell-kollaborativen Systemen besonders wichtig ist, flexibel auf sehr unterschiedliche Arbeitssituationen zu reagieren. Dies erfordert unter Umständen auch einen flexiblen Mechanismus, um Applikationen auf eine zu erledigende Aufgabe zu kontextualisieren. Durch die Komponentisierung von kollaborativen Funktionalitäten und die daraus resultierende Wiederverwendbarkeit können neue höher aggregierte Business-Applikationen zusammengestellt werden. Dabei kann die Zusammenstellung von den Anwendern selbst, die das spezifische Prozesswissen mitbringen, erfolgen. In diesem Fall fällt dem Anwender sowohl die Rolle des Endanwenders als auch die des Anwendungsentwicklers zu. Diese durch den Anwender zusammengestellten Applikationen sind dann abgestimmt auf ihr spezifisches Einsatzgebiet. Damit ist es möglich, Applikationen zu erstellen, die dem jeweiligen Arbeitskontext des Anwenders angepasst sind. Diese Form von zusammengesetzten, kontextualisierten Applikationen wird auch als „Composite-Application“ bezeichnet. Das bedeutet, ein idealtypisches kontextuelles Kollaborationssystem sollte Mechanismen anbieten, die es einem Anwender ermöglichen, Komponenten zu rearrangieren und zu neuen Applikationen zusammenzustellen (Assembly- Funktionalität).

Hupfer et al. leiten aus den Eigenschaften von Komponenten den Vorteil der Wiederverwendbarkeit von Komponenten in kontextuellen Kollaborationssystemen ab. „Finally, contextual collaboration lends itself to reuse of collaborative components. Instead of being entwined in a monolithic collaborative platform, collaborative capabilities (e.g. presence awareness, chat, application sharing) should be designed as

independent, reusable components, with well-defined APIs, that can be embedded and used in any core application.” [Hupfer et al. 2004, S.2]

Griffel stellt bei seiner Definition von Wiederverwendung noch den Aspekt der Bewahrung von Kontextwissen heraus: „Wiederverwendung von Software ist das erneute Anwenden von bei der Entwicklung eines Softwaresystems entstandenen Artefakten und angesammeltem Wissen bei der Entwicklung eines neuen Softwaresystems, um den Aufwand zur Erstellung und Pflege dieses neuen Systems zu reduzieren.“ [Griffel, F.1998 S.16] D.h. das Prozesswissen, welches beim Zusammenstellen von Komponenten angewandt wird, kann in komponentenbasierten Applikationssystemen Wiederverwendung finden. Wenn z.B. ein Anwender eine Applikation erstellt und diese sich als brauchbar für die Bearbeitung eines Geschäftsprozesses erwiesen hat, dann sollte er diese Applikation als Vorlage für ähnliche Geschäftsprozesse wieder verwenden können. Daraus lässt sich folgern, dass es in einem idealtypischen System Mechanismen für die Wiederverwendung von spezifisch zusammengestellten Applikationen geben sollte. Diese wiederverwendbaren Vorlagen werden häufig als Schablonen oder „Templates“ bezeichnet.

Ein System sollte folglich

- Mechanismen anbieten, die es einem Anwender ermöglichen, Komponenten zu Applikationen zusammenzustellen (**Assembly- Funktionalität**)
- Mechanismen für die Wiederverwendung von spezifisch zusammengestellten Applikationen bereitstellen. (**Templating**)

Mit diesen Mechanismen wird die Komponentenzusammenstellung, und damit die Entwicklung neuer Business-Applikationen, von der Hand des Programmierers in die des Anwenders gelegt.

3.3.3.3 Integration auf Objektebene

Eine weitere Alternative zur kontextuellen Integration von kollaborativen Funktionen und Komponenten in bestehende Business Applikationen ist die Integration auf Objektebene. Dabei werden die kollaborativen Funktionalitäten nicht, wie bei der API basierten Integration, an beliebiger Stelle im Code integriert und dem Benutzer an verschiedensten Stellen im UI der Business Applikation zugänglich gemacht. Stattdessen werden diese kollaborativen Funktionen im Kontextmenü spezifischer Objekte, wie z.B. einer Person, eines Dokumentes oder eines anderer Objektes, zur Verfügung gestellt.

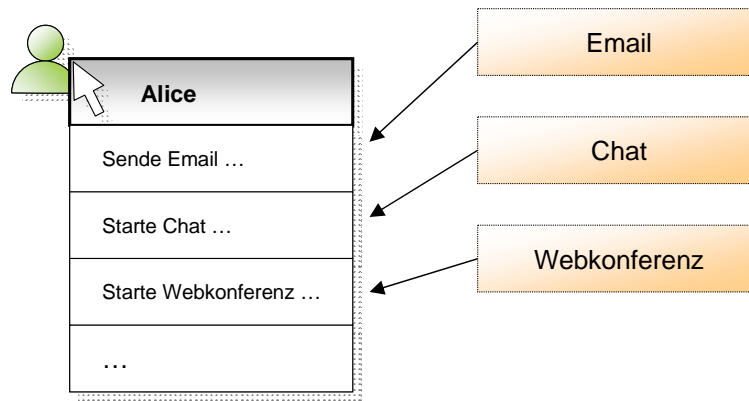


Abb. 12: Integration kollaborativer Funktionen in das Kontextmenü einer Person

Welche Funktionalitäten dabei an welchem Objekt bereitgestellt werden können, hängt vom Typ des Objektes ab. Zum Beispiel können an einem Objekt vom Typ „Person“ folgende kollaborative Funktionen zur Verfügung gestellt werden:

- „Starte Chat...“. Diese Funktion ermöglicht es direkt einen Chat mit der angewählten Person zu starten.
- „Sende E-Mail...“ erlaubt es dem Benutzer eine E-Mail an die angewählte Person zu schicken. Dabei wird z.B. ein Mail-Client mit einer neuen, leeren Mail geöffnet, in der die ausgewählte Person bereits automatisch als Mail Adressat eingetragen ist.
- „Starte E-Meeting...“ startet direkt ein ad-hoc E-Meeting oder ein Screen-Sharing-Session mit der angewählten Person.
- „Plane einen Termin...“ erlaubt dem Benutzer einen Blick in die Free-time-Information des Kalenders der angewählten Person und ermöglicht es so einen gemeinsamen, freien Termin mit der angewählten Person zu finden. Der Benutzer kann dann direkt eine Einladung für den gefundenen Termin an die ausgewählte Person senden.

Neben diesen und anderen kollaborativen Funktionen für eine Person lassen sich auch Funktionen für andere Objekttypen, wie z.B. ein Dokument, bereitstellen, die sich in das Kontextmenü des Objektes integrieren. Für ein Dokument können dies z.B. folgende kollaborative Funktionen sein:

- „Dokument gemeinsam bearbeiten...“ startet ein ad-hoc E-Meeting, in dem das ausgewählte Dokument gemeinsam bearbeitet werden kann. Der Benutzer kann dann ein oder mehrere Personen direkt zu diesem E-Meeting einladen.

- „Sende Dokument...“ erlaubt es dem Benutzer dieses Dokument als Dateianhang an einer E-Mail zu versenden. Diese Option öffnet einen Mail-Client mit einer neuen Mail, in der das Dokument bereits automatisch als Dateianhang zugefügt wurde.
- „Speichern in Activity...“ ermöglicht es dem Benutzer das angewählte Dokument einer seiner aktuellen Activities hinzuzufügen. Dafür wird ihm eine Liste aller aktuell verfügbaren Activities präsentiert, an denen der Benutzer gerade teilnimmt. Das Dokument wird dann der ausgewählten Activity zugefügt und somit auch allen anderen Mitgliedern der Activity zur Verfügung gestellt.

Wie diese Beispiele verdeutlichen, besteht der Wert der Integration nicht ausschließlich, wie in der ersten Stufe beschrieben, darin kollaborative Funktionen oder Komponenten aus einer Business Applikation heraus aufrufen zu können, also z.B. einen Chat Client oder einen Mail-Client zu starten. Der eigentliche Mehrwert für die kontextuelle Kollaboration besteht darin, dass das Objekt, aus dessen Kontext heraus die Funktionalität gestartet wurde, also z.B. die Person oder das Dokument, implizit als Kontextinformation an die kollaborative Komponente übergeben wird und, abhängig von der jeweiligen Funktionalität, semantisch sinnvoll zu dessen Kontextualisierung verwendet wird. Dies bedeutet zum Beispiel, dass die Funktion „Starte Chat...“, die Alice aus dem Kontextmenü der Person „Bob“ auswählt, nicht nur den Chat-Client startet, sondern Bob direkt als Chatpartner auswählt. Alice muss Bob also nicht mehr explizit im Chat-Client als Gesprächspartner suchen und auswählen, da implizit eine Referenz auf die Person „Bob“ als Kontextinformation übergeben wurde. So braucht Alice nur noch ihre Nachricht einzugeben, die dann bei Bob direkt als Text im Chat-Client auftaucht.

Die Integration von kollaborativen Funktionen in die Kontextmenüs von Business-Objekten kann z.B. statisch zum Zeitpunkt der Programmierung der Business-Applikation erfolgen wie im Kapitel 3.3.3.2 erläutert. Damit legt der Programmierer der Business-Applikation statisch fest welche Funktionen im Kontextmenü eines Objektes zur Verfügung stehen.

Ein idealtypisches System kann das Konzept der objektbasierten kontextuellen Integration von kollaborativen Funktionalitäten allerdings flexibilisieren und durch ein generisches Plugin-Framework dynamisch erweiterbar machen. Ein solcher dynamischer Ansatz lässt sich z.B. durch das Plugin und Extension-Point Framework von Eclipse oder durch einen äquivalenten Mechanismus auf einer anderen Plattform realisieren. Diese

dynamische Erweiterbarkeit erlaubt dann auch eine nachträgliche Integration neuer kollaborativer Funktionen in das Kontextmenü eines Business-Objektes, auch wenn diese kollaborative Komponente zum Zeitpunkt der Programmierung der Business-Applikation noch nicht bekannt oder verfügbar war. Wird zum Beispiel später eine Komponente verfügbar, die die Kommunikation mit einer Person via Voice-over-IP (VoIP) erlaubt, so kann diese Funktion dynamisch dem Kontextmenü eines Personenobjektes hinzugefügt werden ohne dazu den Quellcode der Business-Applikation, in der diese Person auftaucht, verändern zu müssen oder auch nur Zugriff darauf zu haben.

3.3.4 Interkomponentenkommunikation

Wenn kollaborative Funktionen als Komponenten realisiert und in eine bestehende Business-Applikation integriert oder im Kontext einer neuen Applikation zusammengestellt werden, so müssen diese Komponenten miteinander interagieren um Kontextinformationen auszutauschen. Ohne eine Komponenteninteraktion würde jede Komponente isoliert von den anderen Komponenten in der Applikation stehen und könnte nicht an den Kontextinformationen teilhaben.

Das in Kapitel 3.3.3 von Griffel beschriebene Merkmal der Interaktion zwischen Softwarekomponenten einer „Composite-Applikation“ wird durch die Interkomponentenkommunikation realisiert. Durch die Kooperation von Komponenten können dann auch neue Funktionalitäten entstehen, die jede einzelne Komponente nicht mitbringt, sondern die dadurch entsteht, dass es Komponenten ermöglicht wird Daten auszutauschen. Herzum definiert vier verschiedene Interaktionsmodi für Komponenten. [vgl. Herzum/Sims 2000, S.210] Diese sind die „integrated interaction“, die „bridged interaction“, die „bus-based interaction“ und die „coordinated interaction“. Diese unterscheiden sich vor allem in dem Grad der Bindung die zwischen 2 Komponenten besteht. Die „coordinated interaction“ z.B. nutzt einen Coordinator oder Broker, der die Komponenten voneinander entkoppelt und ihnen erlaubt miteinander zu kommunizieren, ohne sich kennen zu müssen. [vgl. Herzum/Sims 2000, S.211] Die Integration von Komponenten in bereits bestehende Systeme wird durch die Definition einer Kommunikationsverbindung („Wire“) erreicht. „Da eine Schnittstelle die definierte Grenze einer Komponente darstellt, legt sie auch einen Rahmen für deren Interaktionsfähigkeit fest. Die Frage lautet dann, wie gut ein Interaktionspartner (eine andere Komponente) diese Grenze kennen muss, um

interoperabel (also in gegenseitigen „Verständnis“) kommunizieren zu können.“ [Griffel 1998 S.75]

Motivation dieses Integrationskonzeptes ist es, dass die Komponenten nachträglich vom Anwender integriert werden können. Diese Integration findet daher nicht auf der Programmierenebene statt sondern durch definierte Verbindungen (Wires) mit anderen Komponenten. Durch die Zusammenstellung, Wiring and Konfiguration wird hier die Integration erzielt.

3.3.5 Kontext-Sharing und Shared-Spaces

Um im Team Geschäftsprozesse effizient bearbeiten zu können, ist es oft wichtig, dass mehrere Personen gemeinsam an einer Aufgabe zusammenarbeiten. Dazu ist es wichtig dass Mitarbeiter ihren Arbeitskontext und ihre Kontextinformationen mit Teammitgliedern teilen und gemeinsam nutzen können (Sharing). Es können zwei unterschiedliche Prinzipien des Sharing für die kontextuelle Kollaboration identifiziert werden: Das synchrone Sharing und das asynchrone Sharing.

3.3.5.1 Synchrones Sharing

Synchrones Sharing von Arbeitskontexten findet vor allem Anwendung in der gemeinsamen Ansicht von Arbeitsinhalten zwischen zwei oder mehreren Personen. Z.B. arbeitet Alice gemeinsam mit Bob an einer Präsentation. Dazu führt sie ein Screen-Sharing durch. Das bedeutet, Bob sieht auf seinem Bildschirm genau das, was Alice auf ihrem sieht. [vgl. Use-Case 3.1.13] Allerdings bleibt bei dieser Art des Screen-Sharing die Kontrolle über den Bildschirm bei Alice. Nur sie bestimmt was auf dem Bildschirm zu sehen ist, da nur eine Repräsentation ihres Bildschirms an Bobs Bildschirm übergeben wird (Broadcasting). Diese Art des Sharing ermöglicht es, über einen Arbeitskontext zu kommunizieren und zu kollaborieren ohne dass die beteiligten Personen Aufwand investieren müssen, um den Arbeitskontext für sich herzustellen. Da es sich um eine Bildschirmrepräsentation handelt, ist es mit der Methode des Broadcasting nicht möglich, dass mehrere Personen auch Veränderungen an einem Dokument vornehmen können. Nur derjenige, der das Sharing initiiert hat, kann Veränderungen vornehmen, z.B. an dem gerade gezeigten Dokument. Das Screen-Sharing findet sehr häufig Anwendung in Web-Konferenzen, wenn es wichtig ist, allen beteiligten Konferenzteilnehmern den gleichen Arbeitskontext, über den diskutiert werden soll, zu vermitteln.

Ein weiteres Szenario der gemeinsamen synchronen Bearbeitung von Arbeitskontext wäre, wenn Bob und Alice die Präsentation gemeinsam editieren (**Co-Editing**). D.h. die beteiligten Personen benutzen das gleiche Werkzeug, z.B. Microsoft Powerpoint, um an einem Dokument zu arbeiten. Bei dieser Form der Zusammenarbeit arbeiten alle Beteiligten an dem physisch gleichen Dokument, nicht an einer Kopie. Dadurch können die Personen jeweils sehen, was die anderen Personen schreiben und umgekehrt. Das heißt, Teammitglieder können direkt im Kontext des Dokuments ihre Ideen und Vorstellungen visualisieren, diskutieren und festhalten. Beteiligte Personen bei dieser Art des Sharings sind folglich gleichberechtigt.

3.3.5.2 Asynchrones Sharing durch Shared-Spaces

Wenn Dokumente oder andere Ressourcen gemeinsam bearbeitet werden im Rahmen der Prozesserrfüllung, dies aber nicht in Echtzeit erfolgt wie bei der Methode des Screen-Sharing oder des Co-Editing, wird dies als **asynchrones Sharing** bezeichnet. [vgl. Dourish/Bellotti 1992]

Ein Prinzip des asynchronen Sharing ist, dass ein Anwender ein Dokument bearbeitet und es danach anderen Teammitgliedern zur Bearbeitung bereitstellt. Auf diese Weise werden häufig Geschäftsprozesse und deren Inhalte gemeinsam bearbeitet.

Eine Charakteristik von kollaborationsintensiven Prozessen ist es, dass sie sehr häufig nicht durch ein einzelnes Dokument repräsentiert werden, sondern sie bestehen aus einer Vielzahl von unterschiedlichen Arten von Dokumenten. [vgl. Kapitel 2.2] Die Kollaborationsdokumente, die in unterschiedlichen Systemen oder Applikationen entstehen, verbleiben in der Regel in diesen Applikationen und sind damit entkoppelt von ihrem Kontext, d.h. sie sind zunächst einem Geschäftsprozess nicht eindeutig zuzuordnen. Diese Zuordnung erfolgt häufig als kognitive Wissensleistung der Mitarbeiter. Dieser Problematik wird durch den Einsatz von Shared-Spaces (gemeinsam genutzte Arbeitsumgebung) begegnet.

Shared-Spaces sind geschäftsprozessorientierte virtuelle Arbeitsumgebungen, die es Mitgliedern eines Teams erlauben, alle für einen Geschäftsvorfall relevanten Informationen und die bei der Bearbeitung anfallenden Artefakte zu bündeln, gemeinsam zu nutzen („Sharing“) und auch aus dieser Arbeitsumgebung heraus zu bearbeiten. Das bedeutet, Shared-Spaces können als Kontextcontainer betrachtet werden, in denen sich ein Geschäftsprozess manifestiert.

3. Anforderungen und Konzepte

Damit stellen Shared-Spaces ein Kernmerkmal der kontextuellen Kollaboration dar, insbesondere für die dritte Stufe des vorgestellten Stufenmodells.

Shared-Spaces weisen folgende Charakteristika auf:

- Sie stellen einen **persistenten Platz** dar, der ein oder mehrere Objekte beinhalten kann, welche in der Regel auf asynchrone Weise gemeinsam bearbeitet werden.
- Sie bieten die Möglichkeit **Mitgliedschaften zu definieren**, um so die Personen oder Gruppen von Personen festzulegen, die diese Objekte im Rahmen des Shared-Spaces nutzen können und um ihnen den Zugriff auf Dokumente entsprechend ihrer Rolle zu ermöglichen

Muller et al. beschreiben anschaulich die Probleme, mit denen Anwender konfrontiert werden, wenn sie herkömmliche Kommunikations- und Kollaborationssysteme nutzen, um Dokumente für die gemeinsame asynchrone Bearbeitung bereitzustellen. Jedes Kollaborationswerkzeug adressiert andere Aspekte der Kollaboration. D.h. E-Mail und Instant Messaging sind z.B. Werkzeuge, die ad hoc und kurzzeitig genutzt werden, während z.B. Diskussionsforen und andere gemeinsam genutzte Datenbanken für eine längerfristige Benutzung eingerichtet werden. „The result is often that a single collaborative activity is managed with multiple stores of shared resources and media, such as email, chat, wikis, discussion databases, listservs, and document management systems.” [vgl. Muller et al. 2004, S.1] Muller et al. beschreiben weiter, dass unterschiedliche Arten von Geschäftsprozessen und Aufgaben auch eine unterschiedliche Kollaborationsumgebung für Mitarbeiter verlangt. Es stehen sich dabei folgende Aspekte der Kollaboration gegenüber:

„Small vs large number of participants,

brief vs. extended collaborations,

and informal vs formal structures“ [vgl. Muller et al. 2004, S.1]

Insbesondere der Strukturierungsgrad der Kollaboration spannt ein Kontinuum von eher ad-hoc orientierter, wenig strukturierter Zusammenarbeit auf der einen Seite über semi-strukturierter Kollaboration bis hin zu sehr formaler, prozessorientierter Zusammenarbeit auf der anderen Seite auf.

3. Anforderungen und Konzepte

In Anlehnung an Muller et al. werden im Rahmen dieser Arbeit drei Arten von Shared-Spaces unterschieden:

- Ad-hoc orientierte, Activity-centric-Spaces
- Semi-strukturierte Shared-Workspaces
- sowie eher formale, Workflow getriebene Shared-Spaces.

Diese drei Arten von Spaces werden als geeignet befunden, den unterschiedlichen Anforderungen, die sich aus den verschiedenen Formen und Strukturierungsgraden von Geschäftsprozessen ergeben, gerecht zu werden.

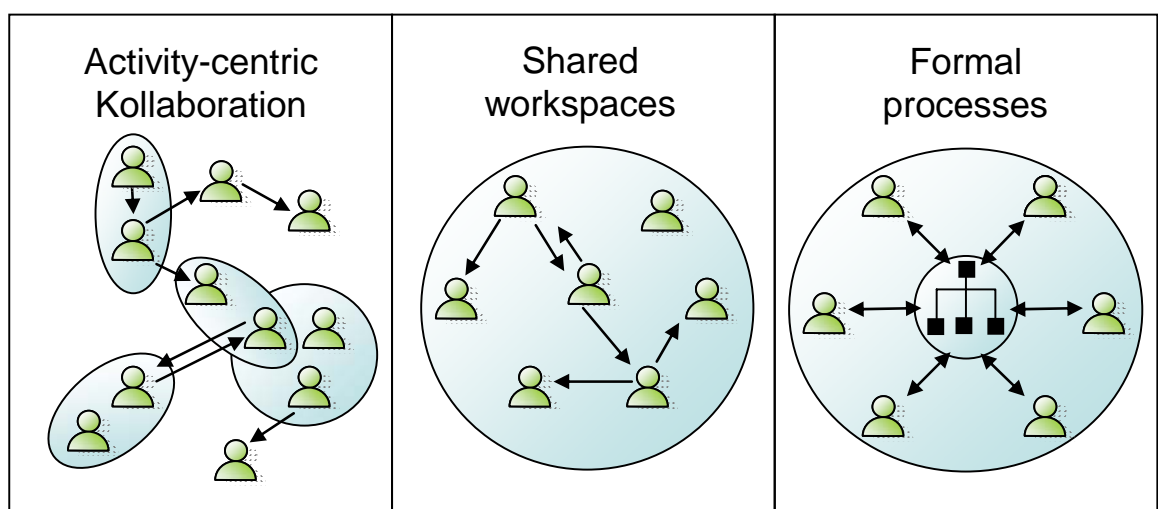


Abb. 13: Kontinuum von ad-hoc zu formalen Spaces
[vgl. Muller et al. 2004 S.1 und Lotusphere 2006 S.74, in Anlehnung]

Ad-hoc-orientierte, Activity-centric-Spaces

Die meisten ad-hoc Kollaborationswerkzeuge, wie E-Mail oder Chat zeichnen sich durch einfache und flexible Handhabung aus. Oft ergeben sich Geschäftsprozesse erst aus einer E-Mail-Anfrage oder einer Chat Kommunikation. Dann allerdings bieten diese herkömmlichen ad-hoc Kollaborationswerkzeuge wenig Möglichkeiten einen Prozess mit mehreren Personen gleichzeitig effizient zu bearbeiten. Um anderen Personen den Kontext der Arbeit zu vermitteln, können zwar z.B. Dokumente an weitere Personen als E-Mailanhang versendet werden, allerdings entstehen bei der gemeinsamen Bearbeitung von Dokumenten Inkonsistenzen und Versionierungsprobleme, da die Anhänge eine Kopie von Dokumenten darstellen.

Activity-centric-Spaces werden im Rahmen dieser Arbeit als eine Form von Shared-Spaces betrachtet, die die Flexibilität und einfache Initiierung von Ad-hoc-Kollaboration bieten und gleichzeitig eine Arbeitsumgebung für eine gemeinsame Nutzung der in den Space eingestellten Objekte darstellen. Durch das Hinzufügen von Objekten zu einer Aktivität wird die Kontextbeziehung einerseits zu der Aktivität und andererseits zu weiteren in der Aktivität befindlichen Objekten hergestellt und damit persistiert. In dem vorgestellten Szenario benutzt Alice einen Activity-centric-Space um ad-hoc mit zwei Mitarbeitern ein spezielles Angebot zu erstellen, welches durch seine Rahmenbedingungen eher als Einzelfall, denn als Routineprozess zu bezeichnen ist. [vgl. Kapitel 3.1.4]. Diese unkomplizierte Vorgehensweise, ad-hoc in kurzer Zeit mit ihren Teammitgliedern gemeinsam ein Angebot auszuarbeiten, kann einen entscheidenden zeitlichen Vorsprung vor Mitbewerbern bedeuten.

Activity-centric-Spaces bieten einen Kontext für Geschäftsprozesse, an denen eine verhältnismäßig kleine Anzahl von Mitarbeitern beteiligt ist, die von kurzer Dauer sind und deren Struktur eher als unformal bezeichnet werden kann. Diese Prozesse sind oft einmalig oder sehr unregelmäßig in ihrem Vorkommen. In einem Activity-centric-Space sind keine formalen Strukturen vorgegeben, die bei der Bearbeitung von Artefakten einzuhalten wären. Inhalte und Strukturen ergeben sich dabei üblicherweise erst interaktiv im Laufe der Bearbeitung des Geschäftsprozesses. Ad-hoc orientierte Spaces sind üblicherweise sehr einfach und schnell vom Anwender zu erstellen, so dass sie durchaus auch zur asynchronen gemeinsamen Bearbeitung eines einzelnen Dokumentes erstellt und von mehreren Personen genutzt werden können. [vgl. zu diesem Kapitel Moran et al. 2005 und Millen et al. 2005]

Semi-strukturierte Shared-Workspaces:

Semi-strukturierte Spaces, oft auch als Workspaces oder Teamspaces bezeichnet, bieten sich als Kontext für Geschäftsprozesse an, die eher wiederkehrend auftreten oder von längerer Dauer oder größerer Komplexität sind, z.B. für größere Projekte. Diese Workspaces bieten oftmals eine Reihe von enthaltenen kollaborativen Komponenten als auch komponentisierten Business-Applikationen. Sie sind häufig als „Composite-Applications“ realisiert. Die Wahl der zur Verfügung gestellten Komponenten gibt damit eine gewisse Struktur vor, in welcher Form welche Inhalte mit welchen Funktionalitäten erstellt und bearbeitet werden können. Die Zusammensetzung der Komponenten in solchen semi-strukturierten Workspaces hat sich üblicherweise aus den Erfahrungen in

der Bearbeitung eines wiederkehrenden Business Prozesses ergeben und stellt somit eine Art „Best Practice“ dar, die bei der Bearbeitung ähnlicher Geschäftsvorfälle einen Produktivitätsgewinn bringen können. Diese Struktur und Zusammenstellung der Komponenten kann oftmals in Form von wieder verwendbaren Templates abgespeichert werden und später als Basis eines neu erstellten Workspaces genutzt werden.

Shared Workspaces lassen den Benutzern genügend Freiheiten, um flexibel auf die Besonderheiten des jeweiligen Geschäftsvorfalles zu reagieren. So erfolgt die Koordination der Mitglieder im Kontext eines solchen Workspace immer noch ad-hoc ohne eingrenzende Vorgaben.

Formale, Workflow getriebene Shared-Spaces

Manche Business Prozesse treten in häufig wiederkehrender und sehr gleichförmiger Form auf. Diese Business Prozesse sind üblicherweise sehr gut strukturierbar und formalisierbar. Eine Kundenbestellung bei einem Internetbuchhändler, wie auch ein Genehmigungsprozess für den internen Einkauf, fallen beispielsweise in diese Kategorie von Business Prozessen. Solche formalen Prozesse können sich z.B. aus unternehmens-internen Richtlinien und Prozeduren ergeben, oder aber auch aus gesetzlichen Auflagen und Verpflichtungen.

Diese Business Prozesse betonen insbesondere den Koordinationsaspekt der Kollaboration. Sie verlangen oft nach einer Regelung welche Mitarbeiter zu welchem Zeitpunkt und unter Einhaltung welcher Geschäftsregeln einen Schritt in diesem Prozess bearbeiten kann oder muss.

Workflows haben sich als Konzept zur Koordination solcher Business Prozesse etabliert. Gemäß der Workflow Management Coalition [WfMC 1999, S.8] ist ein Workflow: „The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.“ Jede Instanz eines solchen Workflows koordiniert damit die Bearbeitung einer oder mehrerer Dokumente und andere Ressourcen durch verschiedene Mitarbeiter. Da die Workflow Instanz als persistenter Container für die zum Business Prozess gehörenden Dokumente fungiert und sie ebenfalls die Mitgliedschaft der Personen regelt, die an dem Workflow mitarbeiten, stellt eine Workflow-Instanz ebenfalls eine Art Shared-Space dar. Die Besonderheit eines solchen Workflow getriebenen Shared-Space ist, dass sich die Liste der Mitglieder und deren

Rollen und Verantwortungen im Verlauf des Prozesses abhängig von bestimmten Geschäftsregeln und vom aktuellen Prozessstatus ändert.

Da kontextuelle Kollaboration als Kollaboration im Kontext eines Geschäftsprozesses definiert ist, stellen Workflows eine wichtige Form der kontextuellen Kollaboration dar. Sie ermöglichen vor allem eine koordinierte Form der kontextuellen Kollaboration.

3.3.6 Das Konzept der Mitgliedschaft (Membership)

Das vorgestellte Szenario zeigt, dass die gemeinsame Benutzung und die Teilhabe an Informationen und Prozessen (engl. sharing) ein wichtiger Aspekt kontextueller Kollaboration ist. Entscheidend dabei ist, dass es in der Hand des Endanwenders liegt, welche Informationen er wann mit wem teilen will. Alice entscheidet sich z.B. im angeführten Szenario eine Präsentation gemeinsam mit Bob zu bearbeiten. [vgl. Use-Case 3.1.13]

Außerdem sind nicht alle Prozesse und Prozessschritte öffentlich, d.h. es gibt Informationen die vertraulich zu behandeln sind, z.B. der interne Prozess der Mitarbeiterbeurteilung am Ende eines Jahres. Diese Informationen sind nur einer bestimmten Gruppe von Personen zugänglich zu machen. Des Weiteren sind auch nicht alle Informationen relevant für jeden. D.h. durch eine Möglichkeit der Informationsfilterung kann geregelt werden, dass die Mitarbeiter vor allem die Informationen erhalten (können), die für den Kontext ihrer Arbeit wichtig sind.

Durch das Konzept der „Mitgliedschaft“ kann diese gemeinsame Nutzung von Dokumenten realisiert werden. Durch die Mitgliedschaft wird festgelegt, welche Personen Teil einer Gemeinschaft (engl. Community) sind oder nicht. Die Mitgliedschaft wird repräsentiert durch eine Liste von Personen bzw. Rollen und bildet damit eine Gruppe von Personen, die an dem gleichen Objekt kollaborieren. Es bildet sich also eine „Gemeinschaft“ um ein Objekt herum. Dadurch wird die Mitgliedschafts-Liste Teil des Objektkontextes.

Die Mitgliedschafts-Liste lässt sich dynamisch anpassen. D.h. der Anwender hat die Möglichkeit, während der Laufzeit neue Mitglieder (Member) hinzuzufügen oder zu entfernen. [vgl. zu diesem Kapitel Silva Filho et al. 2005, S.4]

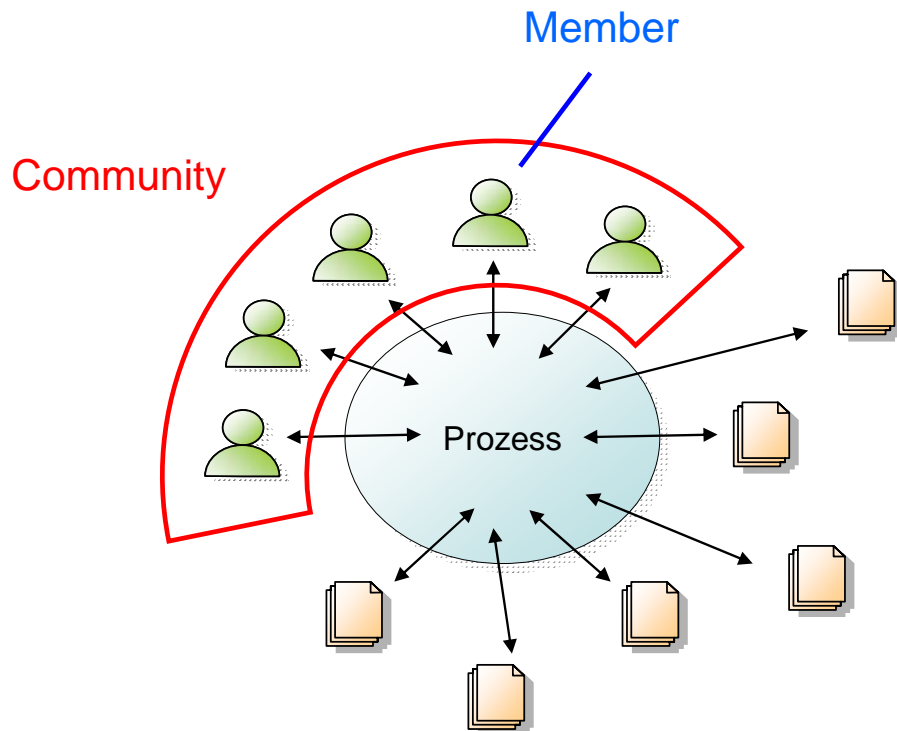


Abb. 14: Community und Member eines Shared-Space

Außerdem folgt das Konzept der Mitgliedschaft dem Rollenkonzept. „Eine Rolle ist eine Abstraktion einer Person, die zur Laufzeit eines Geschäftsprozesses zur Bestimmung einer konkreten Person herangezogen wird.“ [Nastansky et al. 2000 S.295] Für die kontextuelle Kollaboration ist dies ein wichtiger Aspekt, da auf diese Weise Mitglieder unterschiedliche Aktivitäten auf demselben Objekt ausüben können. Ihre Rolle richtet sich damit nach dem Kontext ihrer zu erledigenden Arbeit. Z.B. kann der Inhaber der Rolle „Moderator“ eines Diskussionsforums Beiträge löschen, während der Leser nur Beiträge lesen darf und nicht hinzufügen. Das bedeutet eine Person kann unterschiedliche Rollen innehaben. Diese Rollen bestehen im Kontext des Objekts. Ist die Membership-Liste durch Rollen realisiert, hat das außerdem den Vorteil, dass die Liste definiert werden kann, bevor man die späteren Bearbeiter oder Mitglieder eines Teams kennt. Die Membership-Liste behält ihre Gültigkeit auch bei wechselnden Positionen und Mitgliederzahlen. „Es gelten folgende verbale Regeln:

- Eine Rolle kann mehreren Personen zugeordnet werden.
- Eine Person kann mehrere Rollen innehaben.“ [Nastansky et al. 2000, S.295]

Das Konzept der Mitgliedschaft kann auf unterschiedliche Level von Objekten angewandt werden. Die Mitgliedschaft kann sich auf ein einzelnes Dokument beziehen aber

auch z.B. auf einen Teamspace. Im letzten Fall wird durch die Mitgliedschaft geregelt, wer auf alle im Teamspace befindlichen Informationen und Kollaborations-Funktionalitäten Zugriff hat und auf welche Weise er diese nutzen kann.

3.3.7 Kontext-Awareness

Der englischsprachige Begriff „Awareness“ bedeutet in der direkten Übersetzung „Bewusstsein“ oder „Erkenntnis“. Im Rahmen dieser Arbeit kann „Awareness“ allgemein als „Zustandserkennung“ bezeichnet werden. Awareness-Funktionalitäten zeigen dem Anwender bestimmte Zustände von z.B. Personen oder Dokumenten an, die mit seiner gerade zu erledigenden Arbeit in Beziehung stehen. Kontextuelle Awareness kann demnach z.B. Aufschluss geben, ob Kollegen online sind, an welchen Dokumenten sie arbeiten, mit welchen Devices sie arbeiten, ob sie nicht gestört werden wollen, usw.

In der Literatur lassen sich Hinweise auf verschiedene Formen von Awareness-Funktionalitäten finden. Sie werden den folgenden zwei Kategorien zugeordnet: Der personenbezogenen Awareness (engl. people awareness); sie wird auch als Presence-Awareness bezeichnet. „Presence awareness can also extend beyond just seeing who is online to seeing where contacts are and the processes they are currently involved in“ [Chen et al. 2004, S.5] und der ressourcenbezogenen Awareness (engl. resource awareness). [vgl. Hupfer et al.:2005]

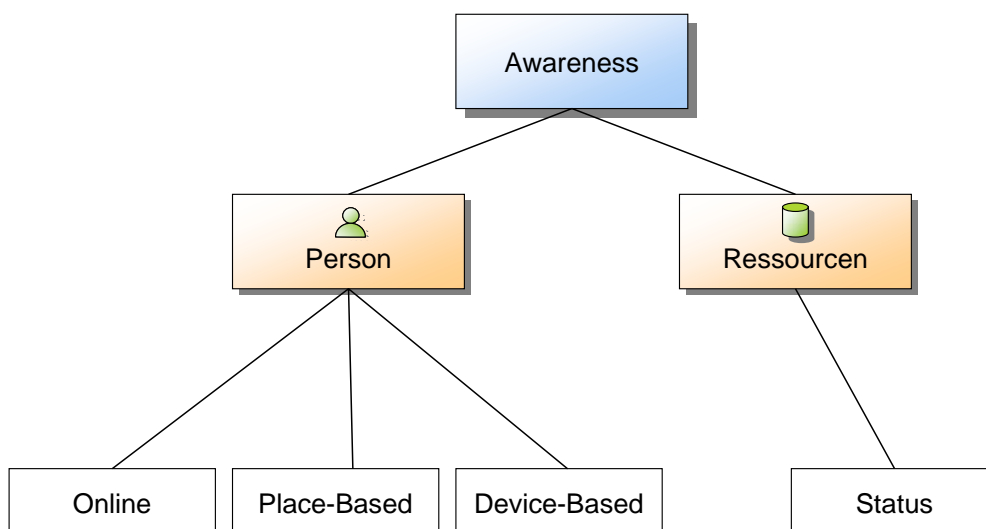


Abb. 15: Formen der Kontext-Awareness

3. Anforderungen und Konzepte

Die People- oder Presence-Awareness dient vor allem der Realtime-Kollaboration. Darunter fällt die Online-Awareness, die Place-based-Awareness, und die Device-based-Awareness. Die personenbezogene Awareness beantwortet dem Anwender damit folgenden Fragenkomplex: Wer ist online? Wo ist jemand? Mit welchem Gerät ist er online? “That is, ordinary presence systems can only provide answers and notifications about person-oriented questions: Who is online?, Is person X online?, and What is person X doing? (to a lesser extent)” [ter Hofte et al. 2002, S4]

Online-Awareness

Die am weitesten verbreitete Funktionalität in am Markt existierenden Realtime-Kollaborationswerkzeugen, ist die Online-Awareness. Sie bietet den Anwendern eine Sicht auf die Präsenz anderer Mitarbeiter im (Kommunikations-)Netz. Online-Awareness bedeutet zunächst nur, zu sehen ob ein möglicher Kommunikationspartner „online“ ist, d.h. ob er am Rechner sitzt und z.B. über Instant Messaging zu erreichen ist. „Turning on the computer and connecting to the internet sets the status to online“ [ter Hofte et al. 2002, S.2] Darüber hinaus haben sich in der letzten Zeit aber auch andere spezifischere Awareness Informationen herausgebildet, da die reine Online-Präsenz wenig aussagt, falls ein Mitarbeiter auch die gesamte Arbeitszeit online sind. So gibt es mittlerweile oft Statusinformationen darüber, ob derjenige ungestört sein möchte, ob er nur kurz abwesend ist, oder ob er telefoniert. „ Other presence information, such as busy, on-the-phone, out-to-lunch, be right back, requires user effort.” [ter Hofte et al. 2002, S.3] Vor allem in der Echtzeit-Kommunikation haben sie sich als nützlich erwiesen, da sie in gewissem Rahmen die Realität abbilden. Mitchell sieht darin eine Verbesserung der Kollaboration von verteilten Teams „ In every day life, we can often alter the perception of others to give them a more informed awareness of our activities, for example by opening or closing our office door, by leaving messages on an answering machine or through gestures. Computer-mediated awareness should support similar facilities.“ [Mitchell 1998, S.2] So gibt es in einigen Applikationen ein Symbol, wenn die Person nicht gestört werden will, z.B. eine geschlossene Tür. Die Tür ist geöffnet dargestellt, wenn derjenige bereit ist. zu kommunizieren. Diese Informationen dienen dem Wunsch, Mitarbeiter möglichst in für sie passenden Momenten zu kontaktieren.

Place-based-Awareness

Place-based-Awareness liefert Informationen darüber, woran Kollegen arbeiten. D.h. wenn ein Mitarbeiter gerade online ist, lässt sich nicht nur sehen, dass er online ist, sondern auch bis zu einem gewissen Grad, woran er gerade arbeitet, also z.B. welche Seite er gerade besucht, welches Dokument er gerade bearbeitet (Falls es ein gemeinsam genutztes (shared) Dokument ist), in welchem Teamspace er sich gerade befindet oder welche Activity er bearbeitet. „ Place-based presence Systems, in addition to the above questions, can also provide answers and notifications about place-oriented questions: Who is here/Who is near? (where here and near may refer to locations, such as a (page on a) website, a file on a shared network drive, etc.), Where is person X?, and What is person X doing there?“ [ter Hofte et al. 2002, S. 3] Mit Nähe ist im Kontext der Place-based-Awareness die virtuelle Nähe gemeint. Von Interesse ist dabei weniger, von jeder beliebigen Person zu erfahren wo sie sich gerade befindet bzw. woran sie gerade arbeitet. Für einen Anwender der sich gerade in einem speziellen Arbeitskontext aufhält, z.B. einen Geschäftsprozess bearbeitet, ist es wichtiger zu wissen, wer sich mit ihm im gleichen Arbeitskontext befindet und was derjenige dort tut. Die Kollaboration und Kommunikation mit Mitarbeitern über diesen gleichen Kontext ist dadurch verbessert. Keiner der im gleichen virtuellen „place“ befindlichen Kollaborationspartner muss erst über den Kontext z.B. einer Anfrage in Kenntnis gesetzt werden. Da sich die Mitarbeiter im gleichen Kontext befinden, kann wirklich ad-hoc kollaboriert werden. Des Weiteren ergeben sich aus der Place-based-Awareness evtl. überhaupt erst die richtigen Kollaborationspartner für einen Anwender. Er erfährt durch diese Art der Awareness, welche Personen für seine Nachfrage überhaupt als Ansprechpartner in Frage kommen.

Das Konzept der „places“ ist dabei ein virtuelles Konstrukt. Ein „Place“ ist nicht beschränkt auf einen physischen Platz (z.B. eine Web-Site oder einen Teamspace) sondern grundsätzlich kann jeder Kontext, in dem sich eine Person befinden kann bzw. an dem sie arbeiten kann, einen „Place“ darstellen also z.B. auch ein einzelnes Dokument eine Activity oder ein E-Meeting. Dabei sind aber gerade die Objekte, die eine Membership haben interessant, da sie durch diese Membership ihre Kontextbeziehung referenziert haben, die sich dann in der Place-based-Awareness nutzbar und sichtbar machen lässt.

Device-based-Awareness

Eine weitere für die Kollaboration interessante Information ergibt sich aus der Device-based-Awareness. Device-based-Awareness Funktionalität kann dem Anwender anzeigen, mit welchem Gerät die möglichen Kollaborationspartner gerade online sind, wie

3. Anforderungen und Konzepte

z.B. mit einem PDA (Personal Digital Assistant), mit einem Handy, oder mit einem Rechner im Büro. Über die reine Anzeige des Gerätes hinaus könnte es auch möglich sein, dass dem Anwender das passende Kollaborationswerkzeug vorgeschlagen oder für ihn ausgewählt wird. Z.B. ist es sinnvoll, einem Kollaborationspartner eher eine SMS (Short Message Service) zu senden als eine Einladung zu einer Video-Konferenz, falls dieser mit einem Handy online ist. Die Kollaboration wird dadurch dem Kontext angepasst. „, ‘Context’ refers to more than just the situation- it means using the appropriate form factor too. Mobile users do not have the time or the display devices to access application and information the way they would when they are sitting at the regular desks with 17-21 in. monitors. Providing the exact collaborative features and information access based on the mobile users’ context - current access device and location - is key.” [Mahowald 2001, S.2]

Die Device-based-Awareness ist damit eine wichtige Funktion der kontextuellen Kollaboration. Das derzeitig benutzte Arbeitsgerät kann als ein Attribut einer Person verstanden werden kann, dass den Arbeitskontext einer Person maßgeblich beeinflussen kann und daher die Kollaborationsoptionen bestimmen kann. Aufgabe kontextueller Kollaboration ist gerade, sich diesem Arbeitskontext anzupassen. Das verschafft beiden Kollaborationspartnern eine optimale Kollaborationsumgebung und sorgt damit für eine höhere Kollaborationsqualität, was auch zu einem verbesserten Prozessergebnis führen kann.

Ressource-Awareness

Neben diesen personenbezogenen Awareness-Informationen ist es aber auch möglich den Anwender über den „Zustand“ der mit seinem Arbeitskontext in Beziehung stehenden Informationsobjekte und Artefakte in Kenntnis zu setzen. In diesem Fall wird von *Ressource-Awareness* gesprochen. Hupfer et. al geben folgendes Beispiel für Ressource-Awareness im Bereich der Softwareentwicklung: „Files in the explorer are decorated with colored icons to signify what teammates are currently doing with the resources (e.g. green indicates a file is currently open and being edited, yellow signals that a file has been modified locally but not checked back into the source control repository, and black denotes that a file has been checked back in).” [Hupfer et al.2004, S.3h

Der Begriff Ressource fasst dabei alle möglichen Informationsobjekte zusammen. Ein einzelnes Dokument kann eine Ressource sein, aber auch ein Meeting, oder eine Aktivi-

tät. Damit können z.B. auch die oben erläuterten Places eine Ressource sein. Grundsätzlich kann jedes Objekt, welches Attribute hat, die sich als Status anzeigen lassen, eine Ressource sein.

Dabei richtet sich die Statusanzeige nach den jeweiligen Attributen, die die unterschiedlichen Ressourcen haben. Für ein Dokument sind z.B. die Statusanzeigen „geöffnet“ und „gerade in Bearbeitung von Person X“, sinnvoll, während für ein Teammeeting die Statusanzeige „wird gerade abgehalten“ interessant für Mitarbeiter ist. Über einen Workflow-Prozess wiederum ist es interessant zu wissen in welchem Workflow Schritt er sich gerade befindet. Durch Icons kann der Status von Attributen der Ressourcen angezeigt werden.

Die unterschiedlichen Statusanzeigen der Awareness-Funktionalitäten können Mitarbeiter in Unternehmen in ihrem Kollaborationsvorgehen unterstützen. Sie dienen daher der Kollaborationskoordination.

Wichtig für die kontextuelle Kollaboration ist, dass diese Awareness-Funktionalitäten als Komponenten zur Verfügung stehen, die in die Business-Applikationen integriert werden. Dadurch ist es für den Anwender möglich, den Status der Ressourcen zu sehen, die für seine gerade im Fokus seiner Aufmerksamkeit stehenden Aufgaben von Bedeutung sind. D.h. er kann den Status der Personen und Ressourcen sehen, die seinen derzeitigen Arbeitskontext bilden.

Die Awareness-Funktionalität schafft durch diese Statusanzeigen für den Anwender ein Mehrwissen über die Geschäftsprozesse, in die er gerade involviert ist. Er kann dadurch schneller und gezielter aus dem Arbeitskontext heraus auf Veränderungen und Probleme reagieren, was die Agilität und Flexibilität der Prozessabläufe verbessern kann.

3.3.8 Notification

Der englischsprachige Begriff „Notification“ bezeichnet Nachrichten, die ein Anwender erhält, falls sich Veränderungen in für ihn relevanten oder interessanten Ressourcen ergeben. Diese Notifications kann er über unterschiedliche Kanäle erhalten, z.B. als Instant Message, als E-Mail, oder auch als SMS auf ein mobiles Endbenutzergerät, wie ein Mobiltelefon oder ein PDA (Personal Digital Assistant).

Die in Kapitel 3.3.7 „Kontext-Awareness“ beschriebenen Awareness-Funktionalitäten dienen dazu, Veränderungen des Arbeitskontextes anzuzeigen, in dem sich ein Anwender gerade befindet. Diese Art der Awareness ist an die Objekte gebunden, die gerade auf dem Bildschirm sichtbar sind. Im Gegensatz dazu dient eine Notification dazu, den Anwender auch dann über wichtige Statusänderungen einer Ressource zu informieren, wenn er sich gerade nicht im Kontext der Ressource befindet. Daher kann die Notification-Funktionalität in der Semantik der kontextuellen Kollaboration auch als Awareness von außerhalb des Fokus liegenden Arbeitskontexten bezeichnet werden. Diese Nachrichten erhält der Anwender unabhängig von dem Kontext, in dem er sich gerade befindet und kann somit Übersicht über verschiedene Arbeitskontexte behalten. Z.B. ist es in der Praxis sehr wahrscheinlich, dass Mitarbeiter Mitglied in mehreren Aktivitäten oder Teamspaces sind, also an verschiedenen Geschäftsprozessen beteiligt sind. Das heißt für den Anwender, solange er sich im Arbeitskontext von Aktivität A befindet, weiß er nicht, ob sich Veränderungen in Aktivität B ergeben haben. Es könnte z.B. möglich sein, dass die Aktivität B nicht weiter geführt werden kann, weil eine Entscheidung von dem Mitarbeiter erwartet wird. Durch die Notification erfährt er über die wichtige Veränderung und kann handeln. Wichtig ist, dass der Anwender keinen Kontextwechsel vollziehen muss, um Veränderungen in anderen Kontexten wahrzunehmen. Erst die eventuell aus der Notification resultierende Handlung erfordert den Wechsel in den anderen Kontext. Übliche Mechanismen zum Versenden von Notifications sind z.B. Mail oder Instant Messages.

News-Feeds stellen eine häufig genutzte Technologie, um als Anwender Nachrichten zu abonnieren. D.h. der Anwender bestimmt selber, von welchen Geschäftsprozessen oder Spaces er über Veränderungen informiert werden möchte. In einem so genannten FeedReader kann er sich die abonnierten Benachrichtigungen anzeigen lassen. Eine der bekanntesten Technologien sind dabei z.B. RSS-Feeds (Really Simple Syndication). Viele Web-Sites bieten mittlerweile ihre Meldungen als RSS-Feed-Abonnement an.

Die Notifications können unterschiedliche Ausprägung haben, insbesondere wenn sie sich auf Inhaltsänderungen beziehen. Sie können entweder nur darüber informieren, dass sich Veränderungen ergeben haben. Meist wird dabei ein Link auf die entsprechende Web- oder Portalseite mit übergeben, oder es wird der veränderte bzw. neue Kontext mit übergeben. [vgl. Silva Filho 2005, S.4-5]

Notifications dienen damit auf unterschiedliche Arten der kontextuellen Kollaboration. Erstens können gerade nicht fokussierte Arbeitskontexte weiterhin auf diese Weise „sichtbar“ (aware) bleiben. Zweitens können sie durch direkte Einbettung von Kontext die zweite Stufe der kontextuellen Kollaboration unterstützen. Die Einbettung des Kontextes erfolgt entweder indirekt als Link oder direkt, indem der veränderte Inhalt als Notification versendet wird. Des Weiteren ist es möglich, als Anwender zu bestimmen, wann, wie und wie oft er über Veränderungen von Kontexten, die außerhalb seines Fokus liegen, informiert wird. Wichtige, dringliche Veränderungen werden z.B. als Instant Message versendet, während Veränderungen in einer Diskussionsdatenbank vielleicht nur einmal am Tag per Mail versendet werden müssen. Das bedeutet, durch die Einflussnahme des Anwenders auf das Wann, Wie und Wie oft kann er die Notifications Kontext entsprechend anpassen. Notifications lassen sich folglich auf den Arbeitskontext kontextualisieren.

3.3.9 Single-Sign-On

Der von Mahowald genannte Vorteil der einfachen Bedienbarkeit eines kontextuellen Kollaborationssystems und der nahtlosen Übergänge zwischen einzelnen Komponenten aus Anwendersicht kann durch den Mechanismus des Single-Sign-On realisiert werden. [vgl. Mahowald 2001, S.2]

Das Contextual Collaboration Framework ist als komponentenbasiertes Framework konzipiert. Jede Komponente wird evtl. mit einem eigenen Sicherheits- und Authentifizierungsmechanismus implementiert. Das bedeutet für den Anwender, dass er, sobald er eine andere Komponente innerhalb eines Systems nutzen möchte, sich gegenüber jeder Komponente authentifizieren müsste. Durch diese Vorgehensweise würde ein Arbeiten ohne Kontextsprünge nicht möglich sein. Der Anwender muss den Kontext seiner Arbeit verlassen, um sich neu zu authentifizieren sobald er z.B. von einer Instant-Messaging-Komponente zu einer E-Mail-Komponente wechseln möchte. Diese Kontextsprünge sollen aber gerade durch kontextuelle Kollaboration vermieden werden. Die reibungslose Benutzung des Systems wird durch die Single-Sign-On-Methode sichergestellt.

Als Single-Sign-On bezeichnet man ein Konzept, durch das ein Benutzer durch einen einmaligen Authentifikationsvorgang (z.B. Passworteingabe) Zugang zu sämtlichen

Komponenten eines Systems bekommt, sofern er die Rechte dazu besitzt. Das heißt, der Benutzer gibt einmalig sein Passwort ein und Aufgabe des Single-Sign-On Mechanismus ist es, den Benutzer gegenüber den anderen Komponenten des Systems zu authentifizieren.

3.3.10 Kontextnavigation

Wie in Kapitel 3.2.4 beschrieben, entstehen durch die Kollaboration mannigfaltige Beziehungen zwischen den einzelnen Dimensionen der Unternehmensorganisation. Zum Beispiel erstellen Personen während der Bearbeitung eines Geschäftsprozesses verschiedene Dokumente die zu diesem Prozess gehören, sie chatten miteinander um ein aufkommendes Problem zu diskutieren oder Referenzen auf externe Dokumente werden zu dem entsprechenden Prozess hinzugefügt etc. So entstehen durch den Kollaborations- und Kommunikationsvorgang verschiedene Beziehungstypen zwischen den unterschiedlichen an der Kollaboration beteiligten Objekten. Die nachfolgende Tabelle zeigt beispielhaft einige dieser Beziehungstypen und der an ihnen beteiligten Objekttypen auf:

Tabelle 2: Objekttypen und Beziehungstypen der kontextuellen Kollaboration

Objekttyp	Beziehungstyp	Objekttyp
Person	Ist Mitglied von	Teamspace
Person	Ist Autor von	Dokument
Person	Hat genehmigt	Dokument
Dokument	Ist referenziert von	Teamspace
Person	Sendet Mail an	Person
Dokument	Ist Kommentar/Antwort zu	Dokument
E-Mail	Gehört zu	Business Prozess
...

Diese Beispiele verdeutlichen, dass es nicht nur einzelne Beziehungen zwischen Objekten verschiedener Typen gibt, wie z.B. zwischen einer Person und einem Teamspace,

3. Anforderungen und Konzepte

sondern dass es auch mehrere verschiedene Beziehungstypen zwischen zwei Objekttypen als auch Beziehungen zwischen Objekten gleichen Typs geben kann, z.B. zwischen zwei Personen oder zwischen zwei Dokumenten.

Diese Beziehungen werden durch das Speichern der Objektreferenzen im Kontext des Geschäftsprozesses explizit und damit auswertbar und analysierbar gemacht.

Diese vielfältigen Beziehungen spannen ein multidimensionales kontextuelles Beziehungsgeflecht zwischen den verschiedenen Objekten im Unternehmen auf. Entlang jeder dieser Beziehungstypen lässt sich ein Schnitt durch das Beziehungsgeflecht ziehen, welcher eine bestimmte Sicht auf die Zusammenhänge im Unternehmen darstellt.

So lässt sich ausgehend von einem bestimmten Objekt, also z.B. der Person „Alice“, über die Beziehung „Ist Autor von“ eine Sicht erzeugen, die alle Dokumente anzeigt die Alice erstellt hat. Ebenso lässt sich z.B. über die Beziehung „Ist Mitglied von“ eine Sicht aller Teamspaces und Activities erzeugen, an denen Alice teilnimmt.

So wird jedes Objekt, welches in den Mittelpunkt der Betrachtung gestellt wird, selbst zu einem Kontext mit dem andere Objekte in einer definierten Beziehung stehen. Nimmt man nun eines der mit dem Kontext in Beziehung stehenden Objekte erneut als Ausgangspunkt einer neuen Sichtenbildung, so navigiert man Schritt für Schritt durch das multidimensionale Beziehungsgeflecht des Unternehmens. So ist Alice z.B. in der Lage sich aus dem Kontext eines Kunden alle zugehörigen Projekt-Teamspaces zu diesem Kunden anzeigen zu lassen um dann von einem der Teamspaces aus die Liste der Mitglieder einzusehen, aus denen sie den Projektmanager identifizieren kann.

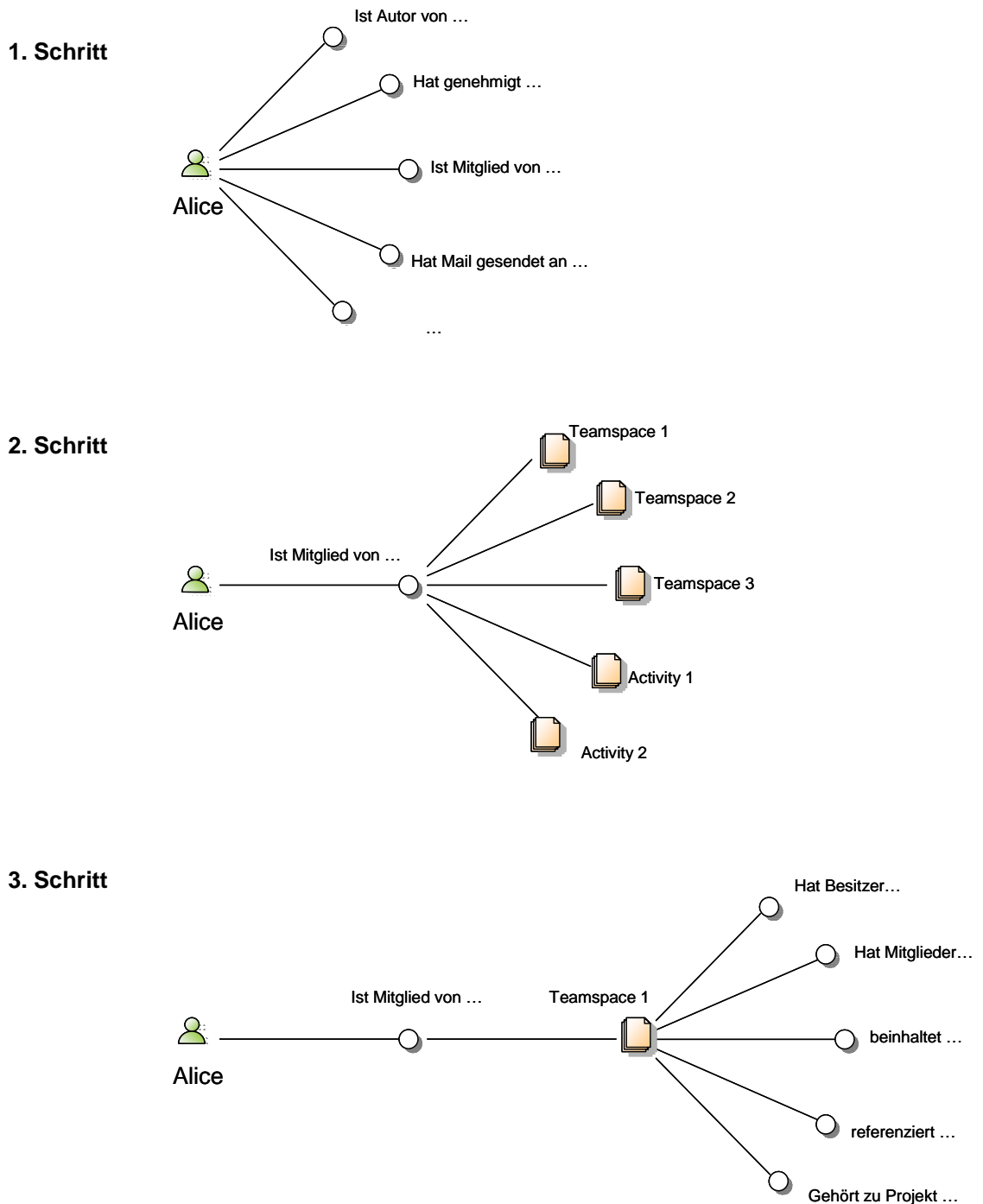


Abb. 16: Schrittweise Navigation über Kontextbeziehungen

So stellt also die Kontextnavigation eine wichtige Methode dar, um das durch die kontextuelle Kollaboration erzeugte Kontextwissen auszuwerten und für den Benutzer transparent und verwertbar zu machen. Durch das Speichern der Beziehungen zwischen verschiedenen an der Kollaboration beteiligten Objekten als Referenzen in dem jeweiligen Kontextobjekt wird dieses Kontextwissen explizit und auswertbar gemacht. Dadurch kann der Benutzer den durch das multidimensionale Beziehungsgeflecht erzeug-

ten Wissensraum systematisch und, durch die Nutzung der Beziehungstypen semantisch sinnvoll, navigieren, um die verschiedensten Aspekte der kontextuellen Beziehungen aus dem Kontext seiner Business-Applikation bzw. seines Business Prozesses heraus zu beleuchten.

Die angefallenen Kontextbeziehungen stellen einen wichtigen Informationspool dar, der neben der Kontextnavigation auch noch zur Erzeugung anderer Sichten und Analysen verwendet werden kann. So kann z.B. durch die Auswahl zweier oder mehrere verschiedener Kontexte eine Schnittmengenansicht erzeugt werden, die es erlaubt alle Personen, Informationsobjekte oder andere mit den Kontexten in Beziehung stehenden Objekte zu identifizieren die entweder direkt oder indirekt mit beiden Kontexten in Beziehung stehen.

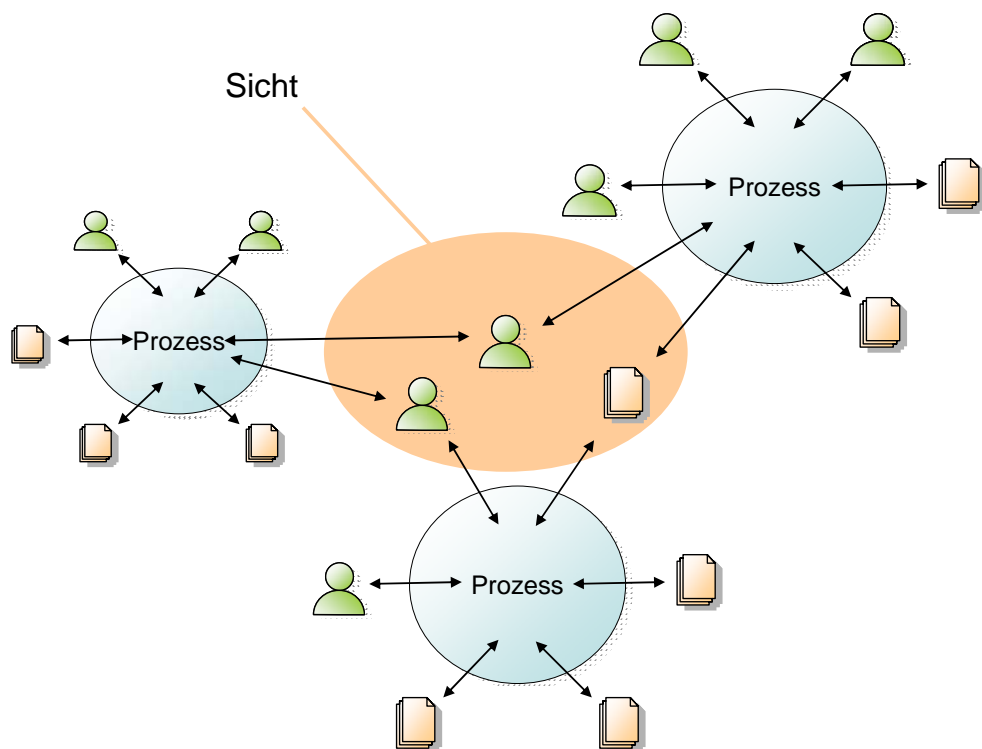


Abb. 17: Sichtenbildung in Kontextbeziehungen

Dies erlaubt z.B. durch die Auswahl mehrerer inhaltlich verwandter Teamspaces die Gruppe von Personen zu identifizieren, die Mitglieder all dieser Teamspaces ist und daher ein engeres virtuelles Team bilden. Andererseits kann z.B. die Art und Häufigkeit der Kommunikationsbeziehungen zwischen mehreren Personen analysiert werden, die z.B. durch Senden von E-Mails oder durch Instant Messaging zu einem bestimmten Kontext oder auch über mehrer Kontexte hinweg in einem intensiven Kollaborationskontakt miteinander stehen. Solche Sichten führen zu Aussagen über die Interaktion

formaler und informeller Gruppen im Kontext von Business Prozessen, die für das Gebiet der Social-Network-Analysis(SNA) interessant sind. Ehrlich und Carboni sehen unter anderem in der SNA eine Möglichkeit strategische Business-Entscheidungen zu unterstützen: „SNAs can help locate expertise, seed new communities of practise develop cross-functional knowledge-sharing, and improve strategic decision-making across leadership teams. [Ehrlich/Carboni 2005, S.1] Diese Aspekte der Auswertung von Kontextbeziehungen und Kontextwissen geht allerdings über den Rahmen dieser Arbeit hinaus und sollen hier nicht im Detail erklärt werden. [Cross etal. 2001]

3.3.11 Metadatenmodell

Das in Kapitel 3.2 vorgestellte Vier-Stufenmodell verdeutlicht, dass der Austausch von kontextuellen Informationen zwischen Komponenten sowie das Persistieren der Referenzen im Kontext der Geschäftsprozesse ein wesentliches Merkmal der kontextuellen Kollaboration ist. Dazu ist es notwendig, dass verschiedene Komponenten diese Kontextinformationen und Referenzen untereinander austauschen und verstehen können. Ein idealtypisches Framework, welches kontextuelle Kollaboration unterstützt, muss deshalb eine Definition von Datenstrukturen bereitstellen, die die grundlegenden, in die Kollaboration involvierten Entitäten wie Businessobjekte, Referenzen auf Businessobjekte, Kontexte etc. einheitlich beschreibt und persistiert. Die Applikationen und Komponenten, die auf diesem Framework aufbauen, sollten dann diese standardisierten Datenstrukturen nutzen, um Kontextinformationen mit beliebigen anderen Komponenten austauschen zu können.

Dazu muss zunächst ein Metadatenmodell für die kontextuelle Kollaboration definiert werden. Das Metadatenmodell beschreibt alle wesentlichen Objekttypen und deren Attribute, die in einem System interagieren. Im Umfeld der kontextuellen Kollaboration sind dies neben den Basiselementen der vier Dimensionen des Unternehmens, also Personen, Geschäftsprozesse, Informationen und Applikationen [vgl. Kapitel 2.2], auch deren spezifischere Untertypen wie Dokumente, E-Mails, Diskussionsbeiträge etc. Jedes dieser Objekttypen hat eine ganz spezifische Menge von Attributen. Eine detaillierte Evaluation und Auflistung aller möglichen Objekttypen und ihrer Attribute ist nicht Gegenstand der Untersuchung im Rahmen dieser Arbeit.

Daher soll hier nur ein kurzer Überblick einiger grundlegender Konzepte eines solchen Metamodells dargestellt werden.

Aus den zuvor genannten Anforderungen und Systemmerkmalen ergibt sich, dass es im Wesentlichen um vier grundlegende Elemente in der kontextuellen Kollaboration geht:

- **Objekte**

Zu diesen gehören alle Objekte die an der Kollaboration beteiligt sind, inklusive Personen, Dokumente, Geschäftsprozesse, etc. Alle Objekte haben zumindest einen Namen, eine ID und einen Typ. Jedes Objekt hat darüber hinaus weitere, typspezifische Attribute. So hat eine Person z.B. ein Attribut „E-Mail Adresse“ während ein Dokument z.B. ein Attribut „Erstellungsdatum“ hat.

- **Referenzen**

Dies sind eindeutige Verweise auf ein spezifisches Objekt. Referenzen werden als Kontextinformation zwischen zwei Komponenten ausgetauscht, wenn z.B. ein Anwender einen Chat aus dem Kontextmenü eines Dokumentes startet. Dann wird an die Chat-Komponente eine Referenz auf den Autor des Dokumentes sowie eine Referenz auf das ausgewählte Dokument übergeben. Komponenten können Referenzen benutzen um auf die Objekte und ihre Attribute zuzugreifen, z.B. um die E-Mail Adresse einer Person zu bestimmen.

- **Beziehungen**

Diese beinhalten alle Beziehungen zwischen zwei Objekten bzw. zwischen einem Objekt und einem Kontext. Beziehungen können unterschiedlichen Typs sein. So ist eine Beziehung vom Typ „ist Mitglied von“ eine Beziehung zwischen einer Person und einem Shared-Space.

- **Kontexte**

Kontexte haben die Möglichkeit alle Beziehungen auf Objekte zu verwalten die mit diesem Kontext in Zusammenhang stehen. Kontexte sind selber Objekte und sind eindeutig referenzierbar. Beispiele für Kontexte sind z.B. Shared-Spaces.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Nachdem im dritten Kapitel verschiedene Anforderungen und Systemmerkmale an ein Framework für kontextuelle Kollaboration identifiziert und im Vier-Stufenmodell systematisiert wurde, werden diese Merkmale nun am Beispiel der *IBM Workplace* Architektur evaluiert. Dazu wird zuerst ein Überblick über die *IBM Workplace* Produktfamilie sowie deren grundlegende Architektur gegeben. Anschließend wird im Kapitel 4.2 geprüft, ob und in welchem Umfang jedes einzelne Strukturmerkmal des idealtypischen Frameworks im Detail in der IBM Workplace Architektur umgesetzt wurde. Dabei wird die Umsetzung jedes Merkmals bewertet bevor am Ende des Kapitels eine tabellarische Zusammenfassung der Ergebnisse folgt.

4.1 IBM Workplace Architektur

Zur Evaluation der *IBM Workplace* Architektur wurde das Produktpaket *IBM Workplace Collaboration Services* Version 2.6 herangezogen. In diesem Paket ist der *IBM Workplace Managed Client* Version 2.6 enthalten. Dieses Produkt wird hier allerdings getrennt von den Collaboration Services beschrieben. Außerdem wurde zur Evaluation des *IBM Workplace* Programmiermodells bzw. der API das *IBM Workplace Software Development Kit (SDK)* Version 2.6 herangezogen. Eine Kurzbeschreibung der weiteren Produkte finden sich im Anhang A.

4.1.1 Überblick über die IBM Workplace Produktfamilie

Der Name *IBM Workplace* steht zunächst für eine Produktfamilie aus dem Bereich der kollaborativen Anwendungen. Damit versteht sich *Workplace* auch eher als Markenname, denn als ein einzelnes Produkt. Unter dem Markennamen werden eine Reihe von einzelnen Produkten und Produktpaketen angeboten. Dazu gehören unter anderem:

- *Workplace Collaboration Services* [vgl. IBM(1) 2006]
 - *IBM Workplace Team Collaboration*
 - *IBM Workplace Collaborative Learning*
 - *IBM Workplace Messaging*

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

- *IBM Workplace Documents*
- *IBM Workplace Web Content Management*
- *IBM Workplace Managed Client*
- *Workplace Services Express*
- *Workplace for Business Controls and Reporting*
- *Workplace for Business Strategy Execution*
- *Workplace Forms*
- *Workplace Dashboard*

Außerdem werden für die Entwicklung von *Workplace*-Komponenten noch

- *Workplace Designer*
- *Workplace Software Development Kit 2.6*

bereit gestellt.

Workplace Collaboration Services ist ein Paket von mehreren Teamworkfunktionen, die selbst wieder als Offering-Pakete angeboten werden. Diese sind: *IBM Workplace Team Collaboration*, *IBM Workplace Collaborative Learning*, *IBM Workplace Messaging*, *IBM Workplace Documents*, *IBM Workplace Web Content Management*. Darüber hinaus beinhaltet dieses Paket noch den *IBM Workplace Managed Client*, allerdings ist hierfür eine separate Lizenz erforderlich. Außerdem sind die allgemeinen Funktionen Instant Messaging und Presence-Awareness, sowie Verzeichnissuche und People Finder in vielen *Workplace Collaboration Services* Kontexten verfügbar.

Das Offering-Paket ***IBM Workplace Team Collaboration*** beinhaltet die Funktionen Applications, Web-Conferences und Templates. Der Anwender hat Zugriff auf die *Workplace*-Anwendungen „Team Spaces“ und „Documents“, auf Domino Anwendungen mit HTML-Unterstützung und auf kundenspezifische Anwendungen. „Web-Conferences“ ist ein weiteres Element, das spezifische Funktionalitäten zur Online-Präsentation und E-Meetings und deren Moderation bereithält. In „Templates“ kann der Anwender eigene Templates erstellen, anpassen und verwalten.

Das Offering-Paket ***IBM Workplace Collaborative Learning*** bietet Funktionen die für Kursteilnehmer den Zugriff, die Verwaltung und Anzeige von Online-Kursen ermögli-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

chen. Für Kursentwickler und Kursleiter bietet es die Möglichkeit der Verwaltung von Ressourcen und Lernprogrammen. Außerdem gibt es Autorenwerkzeuge, die die Entwicklung von Kursen unterstützen.

IBM Workplace Messaging beinhaltet die Funktionen Mail, Kalender und Personal Adressbook. Für den *IBM Workplace Managed Client* bietet es außerdem noch eine Offline Unterstützung und das integrierte Instant Messaging.

IBM Workplace Documents bietet systematischen, kontrollierten Zugriff auf Dokumente, Funktionen zur Dokumentenverwaltung mit integrierten Teamworkfunktionen. Es unterstützt sowohl den Browser als auch den WMC. *IBM Workplace Documents* bietet für den Browser außerdem offline-Unterstützung.

IBM Workplace Web Content Management beinhaltet Funktionen zum Content Authoring, zur Versionssteuerung und Rollback, zur Integration von Informationen von verschiedenen Quellen, zur personalisierten Darstellung und zur Unterstützung für mehrere Datenbanken.

IBM Workplace Managed Client bietet den Zugriff auf kollaborative Funktionalitäten in Form einer Rich-Client Desktop-Applikation. Zu diesen kollaborativen Funktionalitäten gehören:

- **Workplace Messaging.** Der *Workplace Managed Client* stellt eine Rich Client Benutzeroberfläche für den Zugriff auf die o.g. Messaging Funktionalität der *Workplace Collaborative Services* zur Verfügung.
- **Lotus Notes Plugin** erlaubt den transparenten Zugriff auf den *Notes Client* und alle *Notes/Domino* basierten Anwendungen
- **Der IBM Activity Explorer** dient der Verwaltung und Übersicht von Aktivitäten, die einem Projekt angehören. Der Activity-Explorer stellt dafür Funktionen der synchronen und asynchronen Kollaboration zur Verfügung. Er steht nur im *Workplace Manged Client* zur Verfügung, nicht aber im Portal basierten Browser UI.
- **Workplace Documents** stellt ebenfalls eine Rich Client Oberfläche für den Zugriff auf die bereits genannte Dokuments Funktionalität
- Das **IBM Data Access Tool** dient der Erstellung von relationalen Datenbankwendungen

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

- Ein **integrierter Browser** zum Öffnen und Navigieren von Webseiten direkt aus dem Managed Client.
- Zugriff auf die Web basierten **Workplace Applications und Teamspaces** durch den integrierten Browser

4.1.2 Überblick über die IBM Workplace Architektur

Die *IBM Workplace* Plattform versteht sich als Plattform für kollaborative Enterprise Software. Sie basiert auf dem *IBM WebSphere Application Server* und auf *IBM WebSphere Portal*. Die Abb. 18 gibt eine schematische Übersicht über die *Workplace* Architektur.

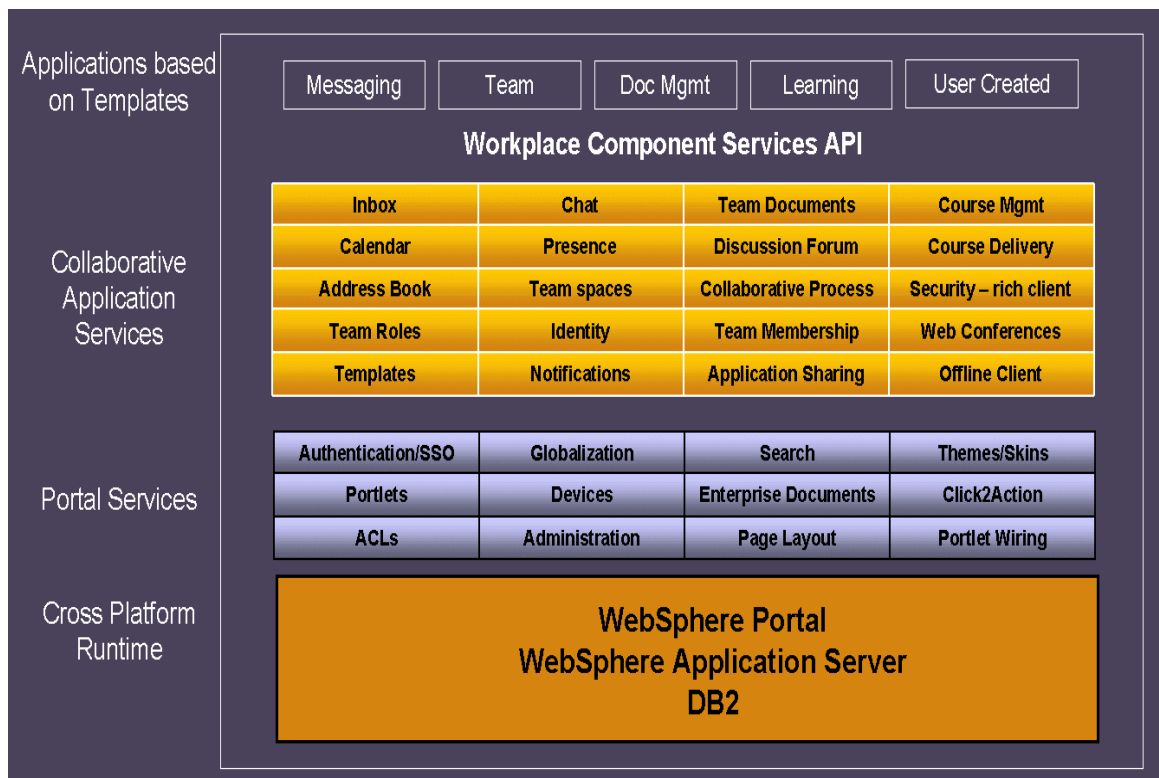


Abb. 18: Workplace Architektur

[Monson et al. 2005 S.44]

Das Schaubild zeigt die unterschiedlichen Schichten der *IBM Workplace* Architektur, die im Wesentlichen auch die Strukturmerkmale von Portalen aufweist, die in Kapitel 2.4 „Grundlagen der Portaltechnologie“ vorgestellt wurden. *Workplace* bietet eine Reihe von standardbasierten kollaborativen Applikationen, die hier als „Collaborative Ap-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

plikation Services“-Schicht dargestellt ist. Diese kollaborativen Applikationen lassen sich über die *Workplace Component Services API* nutzen. Darauf aufbauend lassen sich höher aggregierte Applikationen als Komponenten in bestehende Systeme integrieren, die diese kollaborativen Services nutzen. Im Schaubild werden diese als oberste Schicht dargestellt. Grundsätzlich ist *IBM Workplace* zunächst eine webbasierte Portalarchitektur und die Applikationen werden in der Regel als Portalseiten im Browser angezeigt, die sich aus einzelnen Portlets zusammensetzen. *Workplace* bietet dabei die Möglichkeit solche Applikationen Template basiert zusammenzustellen.

Eine Besonderheit stellt der *IBM Workplace Managed Client* dar. Er lässt, neben dieser browserbasierten Sicht auf *Workplace* Applikationen durch den integrierten Browser, auch die Sicht auf Desktop basierte Applikationen, die Plugins, zu. Plugins sind auf dem Eclipsestandard basierende Komponenten für den *IBM Workplace Managed Client*, die sich in *IBM Workplace* Systeme integrieren lassen. Dabei ist eine Besonderheit des *IBM Workplace Managed Client*, dass diese Plugins serverseitig installiert und gewartet werden können. Der *Workplace* Server organisiert die Übertragung der Plugins an die einzelnen Clients. Daher können für den *Workplace Managed Client* Offline-Funktionalitäten bereitgestellt werden. Der *Workplace Managed Client* vereint damit die Vorteile einer Portalarchitektur mit denen der Client-basierten Architektur. D.h. *IBM Workplace* bietet somit einerseits die Möglichkeit neue Komponenten zentral hinzuzufügen und zu warten und gleichzeitig die Benutzungsfreundlichkeit und die Offline-Funktionalitäten einer Desktop-Applikation, um dem Anwender eine flexible Arbeitsumgebung zu bieten. [vgl Bargas 2005] und [IBM(1) 2006] und [Groeger/Gorman 2004]

4.2 Umsetzung der Anforderungen

In diesem Kapitel wird die Umsetzung der im dritten Kapitel definierten Anforderungen und Strukturmerkmale des idealtypischen Frameworks für kontextuelle Kollaboration am Beispiel der *IBM Workplace* Architektur evaluiert. Dabei steht im Vordergrund ob und in welchem Umfang die *Workplace* Architektur die genannten Konzepte und Strukturmerkmale implementiert und in Form eines Frameworks den Anwendern und Entwicklern zur Verfügung stellt. Der Fokus liegt dabei darauf, zu prüfen, in welchem Rahmen die *Workplace* Architektur geeignet ist, kontextuell-kollaborative Systeme zu entwickeln und kollaborative Funktionalitäten in bestehende Applikationen zu integrieren oder zu neuen Applikationen zusammenzustellen.

Es ist hingegen nicht das Ziel dieser Arbeit die verschiedenen kollaborativen Funktionen des *IBM Workplace* Systems, wie z.B. die E-Mail-Funktionalität oder die Web-Konferenzen, zu evaluieren.

4.2.1 Komponentisierung

Das Komponentenmodell wird in der *IBM Workplace*-Architektur durch Portlets realisiert. Die in Kapitel 2.4 beschriebenen Basis-Elemente von Portalen liegen auch der *IBM Workplace* Architektur zugrunde. Das bedeutet, Portlets sind die grundlegende Komponenteneinheit in der *IBM Workplace* Architektur. „They [portlets] are reusable components that provide access to applications, Web-based content, and other resources. Web pages, services, applications, and syndicated-content feeds can be accessed through portlets.” [Iyengar et al. 2005, S.6] Da die *IBM Workplace* Portalarchitektur durch *IBM WebSphere Portal* realisiert wird, entsprechen die Portlets dem *WebSphere Portal*-Standard. *WebSphere Portal* unterstützt dabei die Standard Portlet API (JSR-168). Komponenten, die auf diesem Standard basierend entwickelt wurden, können in bestehende Systeme, die auf der *Workplace* bzw. einer *WebSphere Portal* Architektur basieren, integriert werden.

Spezielle *Workplace* Komponenten erweitern das Komponentenmodell von Portlets mit einer so genannten „Collaborative Component“ [Groeger/Gorman 2004]. Das IBM Software Development Kit (SDK) beinhaltet eine Reihe von dokumentierten Schnittstellen, die es ermöglichen eigene „Collaborative Components“ zu implementieren.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Diese können in eine kollaborative Infrastruktur eines *IBM Workplace*-basierten Systems integriert werden. Diese Interfaces werden von IBM als „Collaborative Application Component Interfaces“ bezeichnet. Die Collaborative Components implementieren spezifische *Workplace* Component interfaces die es ihr ermöglichen in *Workplace* Applikationen und Templates zu integrieren und deren Services zu nutzen. Diese *Workplace* Components haben die Fähigkeit sich auf die Instanz einer *Workplace* Applikation zu kontextualisieren, d.h. dass sie am Lebenszyklus (engl. Lifecycle) der Applikation teilnimmt und ihre Daten lokal im Gültigkeitsbereich der Applikation verwalten kann. Dagegen sind Portlets immer global. Der Unterschied soll an einem Beispiel verdeutlicht werden:

Ist ein elektronisches Pinboard (engl. Noteboard) als *Workplace*-Komponente implementiert, dann nimmt sie am Lifecycle der Applikation teil d.h. wird eine *Workplace* Applikationsinstanz aus einem Template neu angelegt so werden alle in dieser Applikation liegenden *Workplace*-Komponenten inklusive des Noteboards darüber informiert. Die Applikation ruft eine Methode der *Workplace* Komponente auf, die es der Komponente erlaubt sich zu initialisieren. Dabei hat sie Zugriff auf die Instanz ID der neu angelegten Applikation d.h. das Noteboard legt einen neuen Datenbereich in seinem Data Store speziell für diese Applikationsinstanz an. Während des Lebenszyklus der Applikation werden dann alle Noteboardbeiträge (notes), die in dieser Komponente erstellt werden in diesem gesonderten Datenbereich gespeichert. Dies erlaubt der Komponente nur die Beiträge anzuzeigen die in der entsprechenden Applikationsinstanz angelegt wurden. Beiträge die aus anderen Applikationsinstanzen entstammen werden explizit nicht berücksichtigt. Wird die Applikation zu einem späteren Zeitpunkt gelöscht, so wird die Noteboard-Komponente benachrichtigt, die daraufhin alle zu dieser Applikationsinstanz gehörenden Notes aus ihrem Data-Store löscht. Dadurch verhalten sich die *Workplace*-Komponenten und die in ihnen enthaltenen Daten wie ein echter Bestandteil der *Workplace*-Applikation, obwohl Komponenten ihre Daten verteilt in den verschiedensten Backendsystemen halten können.[vgl. Groeger/Gorman 2004]

Im Gegensatz dazu hat ein Noteboard, welches als reguläres Portlet implementiert ist, keinen Zugriff auf Informationen über die *Workplace*-Applikationsinstanz in der sie sich befindet. Daher fehlt in diesem Fall dem Portlet die Information um sich auf die Applikationsinstanz zu kontextualisieren. Das Portlet kann die Daten nur global anzeigen, d.h. Notes die in einem Noteboard in einer Applikationsinstanz eingegeben wurden,

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

sind auch in einer anderen Applikationsinstanz sichtbar. Es gibt einen zentralen Datenbestand, in dem alle zu dem Portlet gehörenden Notes gespeichert sind.

Zusammenfassend lässt sich sagen, dass *IBM Workplace* damit ein Komponentenmodell definiert, welches die verschiedenen Voraussetzungen für kontextuelle Kollaboration, wie in Kapitel 3.3.2 definiert, erfüllt. Eine detailliertere Bewertung der Komponentisierung in *Workplace* wird zusammen mit der Komponentenintegration am Ende des nächsten Kapitels geliefert.

4.2.2 Komponentenintegration

Die Bedeutung der Möglichkeiten kollaborative Funktionen in ein bestehendes kontextuell-kollaboratives System integrieren zu können, wurde in Kapitel 3.3.3 hinreichend erläutert. In diesem Abschnitt werden die Evaluationsergebnisse in den folgenden zwei Bereichen vorgestellt:

- Es wird hier evaluiert auf welche Weise Anwender/Programmierer die Funktionalität der Integration nutzen können.
- Es wird evaluiert, inwiefern IBM die Funktionalität für die bereits implementierten Services (IBM Collaboration Services) nutzt.

4.2.2.1 API basierte Integration

IBM Workplace bietet Möglichkeiten kollaborative Funktionalitäten durch eine API basierte Integration zu einem bestehenden *IBM Workplace*-basierten System hinzuzufügen. Diese Integration kann durch die dokumentierten APIs und SPIs, welche durch das *IBM Software Development Kit (SDK)* zur Verfügung stehen, realisiert werden.

Die von *Workplace* implementierten und in bereits existierenden *Workplace*-Komponenten genutzten kollaborativen Funktionalitäten können durch spezielle APIs, die „Component and Application Infrastructure Services APIs“ angesprochen werden. Diese dokumentierten Funktionalitäten können daher genutzt werden um eigene Komponenten mit diesen Funktionalitäten anzureichern. [vgl. IBM e 2006]

Bewertung

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Diese Art der Integration von kollaborativen Funktionalitäten ist für die kontextuelle Kollaboration eine wichtige Voraussetzung um Business-Applikationskomponenten mit diesen Funktionen anzureichern. Durch diese Integrationsmöglichkeit kann dem Hauptanliegen der kontextuellen Kollaboration, dem Kollaborieren aus dem Kontext von Business-Applikationen, Rechnung getragen werden. Die API basierte Integration geht dabei mit den in Kapitel 3.3.3.1 erläuterten Vor- und Nachteilen einher. Das heißt diese Art der Integration ist generell nur anwendbar für selbst entwickelte Komponenten. Es lassen sich nachträglich keine bestehenden Komponenten auf diese Weise mit kollaborativen Funktionalitäten anreichern.

4.2.2.2 Komponentenbasierte Integration

Die komponentenbasierte Integration wird in der *IBM Workplace* Architektur durch das bereits in Kapitel 3.3.3.2 erläuterte Konzept der „Composite-Applications“ realisiert. *IBM Workplace* bietet durch dieses Konzept die Möglichkeit, einzelne Komponenten, Portlets oder Plugins, zu höherwertigen Applikationen zu aggregieren. Ein Beispiel ist die Zusammenstellung einer speziellen Applikation zur Durchführung eines Marketing-Projektes, in dem neben einigen kollaborativen Komponenten, wie z.B. einem Teamkalendarer, auch eine spezifische grafikverarbeitende Komponente benötigt wird. Solch eine aus mehreren Komponenten zusammengestellte Applikation wird dann „Composite-Application“ genannt.

Für die Zusammenstellung von Portlets durch einen Anwender stellt *IBM Workplace* den *IBM Workplace Builder* als Werkzeug zur Verfügung. Bei der Benutzung des *Workplace Builder* wird dem Anwender eine Liste mit auswählbaren Komponenten angezeigt.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Abb. 19: Anlegen eines neuen Teamspace

Name	Owner	Category	Modified
Blank Template	admin	None	December 22, 2005 6:49:18 AM GMT+01:00
Chat Room	admin	Team Space	December 22, 2005 6:49:18 AM GMT+01:00
Course Collaboration	admin	Learning	December 22, 2005 6:49:20 AM GMT+01:00
Customer Support Team	admin	Team Space	December 22, 2005 6:49:18 AM GMT+01:00
Database	admin	Workplace Application	February 6, 2006 6:23:54 PM GMT+01:00
Discussion	admin	Team Space	December 22, 2005 6:49:18 AM GMT+01:00
Document Library	admin	Document Library	December 22, 2005 6:49:19 AM GMT+01:00
Domino Application Access	admin	Workplace Application	December 22, 2005 6:49:20 AM GMT+01:00
Employee Human Resources Site	admin	Team Space	December 22, 2005 6:49:19 AM GMT+01:00
Event Planning	admin	Team Space	December 22, 2005 6:49:19 AM GMT+01:00

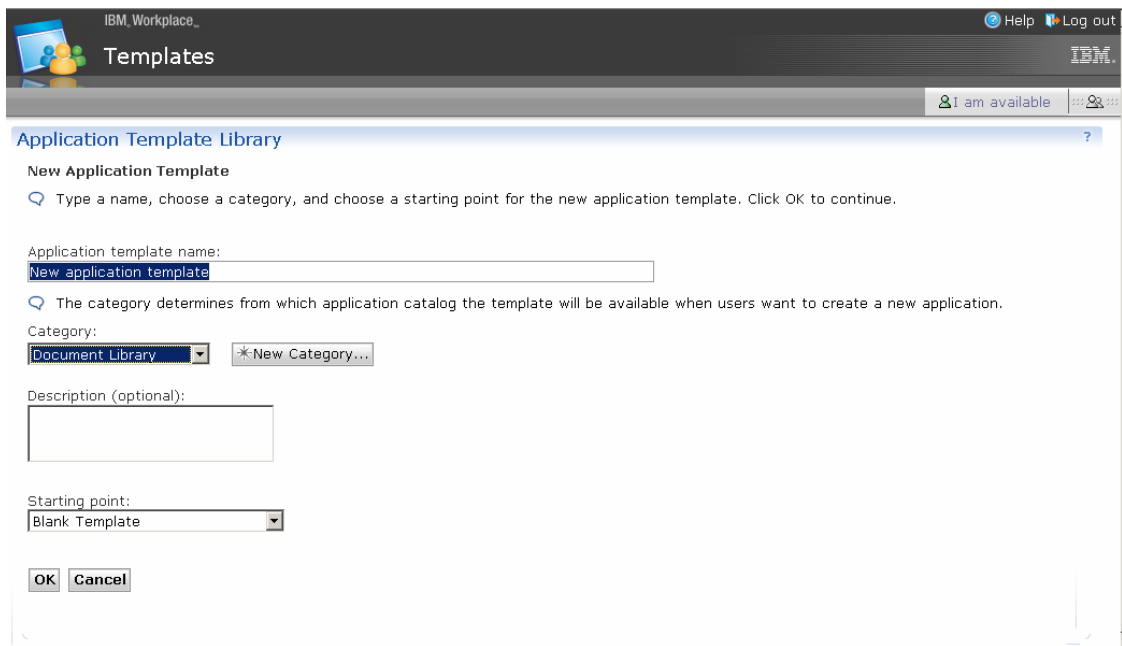
Abb. 20: IBM Workplace Template Library

Der Anwender hat verschiedene Möglichkeiten diese neuen Applikationen zu erstellen. Er kann eine Applikationsinstanz erstellen basierend auf einem Applikations-Template. Dafür steht eine Template-Library zur Verfügung aus der er Templates auswählen kann. Des Weiteren kann er eine Composite-Applikationen zusammenstellen und diese als Template definieren oder aber er kann eine neue Composite-Application zusammenstellen. Sowohl bei der der Erstellung von Templates als auch einer Applikationsinstanz gibt es Parameter (Points of Variability (POV)), an denen Veränderungen und Anpassun-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

gen an den einzelnen Komponenten vorgenommen werden können. Diese Anpassungen betreffen die allgemeinen Properties wie z.B. den Namen der Komponente, das Layout sowie die Rollen, die sich für eine Komponente definieren lassen. Darüber hinaus lassen sich für die Templates auch Standardwerte festlegen. Eine erstellte Composite-Application-Instanz ist dann sichtbar unter dem Menüpunkt „Workplace Applikationen“.

Die in *IBM Workplace* Teamspaces stellen eine spezielle Form der Composite-Application dar. Teamspaces lassen sich genau nach diesem Prinzip zusammenstellen.



The screenshot shows the 'IBM Workplace' interface with a 'Templates' window. The window title is 'Application Template Library'. Below the title, there is a section for 'New Application Template'. A help icon and text state: 'Type a name, choose a category, and choose a starting point for the new application template. Click OK to continue.' The form contains the following fields:

- 'Application template name:' with a text input field containing 'New application template'.
- 'Category:' with a dropdown menu showing 'Document Library' and a button '*New Category...'.
- 'Description (optional):' with an empty text area.
- 'Starting point:' with a dropdown menu showing 'Blank Template'.

At the bottom of the form are 'OK' and 'Cancel' buttons.

Abb. 21: Anlegen eines neuen Templates

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

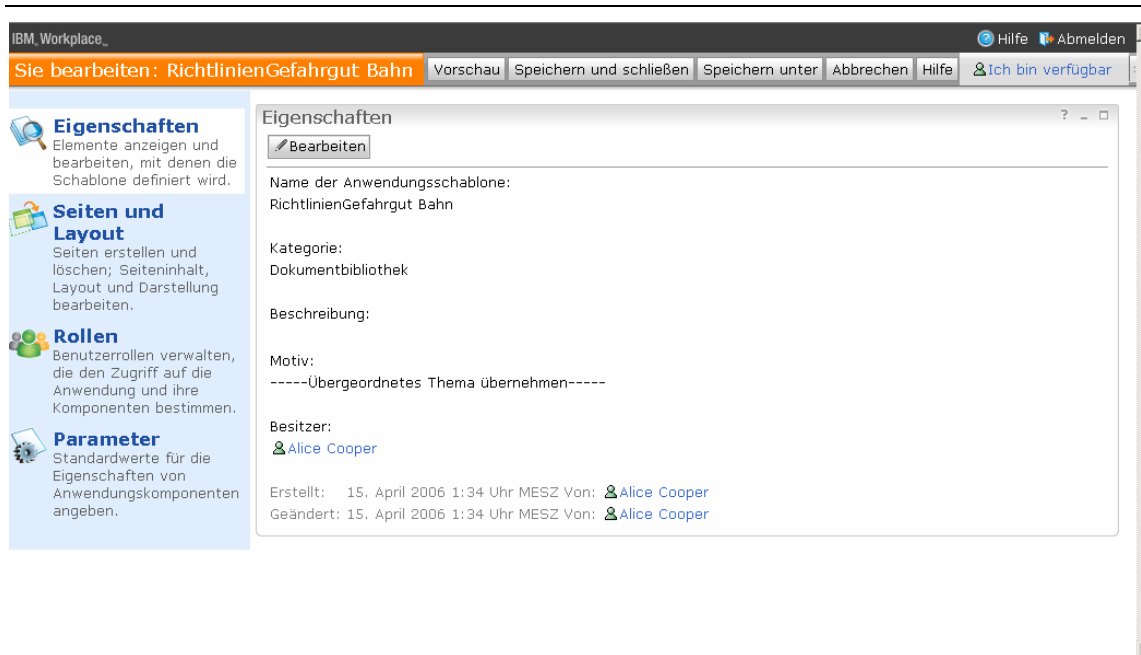


Abb. 22: IBM Workplace Builder – Eigenschaften und Points of Variability

Bewertung für die Strukturmerkmale Komponentisierung und Komponentenintegration

Die Evaluierung des Kriteriums der **Komponentisierung** hat gezeigt, dass grundsätzlich die Forderung der Komponentisierung für kontextuelle Kollaborationssysteme durch die *IBM Workplace* Architektur erfüllt ist. Dies gilt sowohl für die Portlet-Komponenten als auch für Plugin Komponenten.

In Kapitel drei wurde als ein wichtiges Kriterium für Komponenten ihre Repräsentation durch APIs und Interfaces nach außen genannt. Das bereits hier Erwähnung gefundene *IBM Workplace Software Development Kit* beschreibt und dokumentiert diese APIs und Interfaces für eine Reihe von kollaborativen Komponenten, z.B. für eine Web-Konferenz-Komponente.

Für Softwareentwickler, die Business-Applikationen auf Basis der *IBM Workplace* Plattform entwickeln, ist es durch die API-Repräsentation möglich, diese eigenen Applikationen mit den von *IBM Workplace* angebotenen kollaborativen Funktionalitäten anzureichern. Das bedeutet *Workplace*-Komponenten lassen sich, sofern sie dokumentiert sind, zur Entwicklungszeit integrieren. Folglich ist so auch das Kriterium der Integrationsfähigkeit der Architektur auf API-Ebene erfüllt. Da es im Rahmen dieser Arbeit Aufgabe ist, die Struktur der *IBM Workplace* Architektur auf ihre Fähigkeit hin zu un-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

tersuchen, als kontextuell-kollaborative Plattform zu dienen, wurden nicht die APIs einzelner kollaborativer *Workplace* Komponenten evaluiert. Es kann aber gesagt werden, dass mit dem SDK zwar einige aber nicht alle Komponenten vollständig veröffentlicht sind. Ein Beispiel hierfür sind die „Activities“.

Die Anforderung der komponentenbasierten Integration als Strukturmerkmal wird in *Workplace* durch das Konzept der „Composite-Application“ erfüllt. Der *IBM Workplace-Builder* als Werkzeug, das es dem Endanwender ermöglicht portalbasierte Applikationen einfach und flexibel zusammenzustellen, kann als der in Kapitel 3.3.3.2 geforderte *Mechanismus für die Zusammenstellung durch den Endanwender* angesehen werden. Darüber hinaus können die hier beschriebenen Möglichkeiten, Applikationen als Templates zu definieren als *ein Mechanismus für die Wiederverwendung von spezifisch zusammengestellten Applikationen* gelten.

Der *Workplace-Builder* als einfach zu bedienendes Werkzeug findet derzeit im *Workplace Managed Client* noch keine Entsprechung. Das bedeutet, es lassen sich zwar grundsätzlich Plugins zu einer höherwertigen Applikation zusammenstellen, allerdings wird dies in der Regel programmierseitig realisiert und lässt sich selten durch den Endanwender durchführen.

Die Kontextualisierung auf spezielle Geschäftsprozesse wird durch Komponentenmodell der *IBM Workplace* Plattform auf unterschiedlichen Ebenen realisiert. Die grundsätzliche mögliche Auswahl an Komponenten kann schon Spezialisierung z.B. auf eine Geschäftsbranche bedeuten. Die Zusammenstellung einzelner Komponenten zu „Composite-Applications“ kann eine weitere Anpassung darstellen. Die POV können die Kontextualisierung auf einen spezifischen Prozess noch verbessern.

Ein positiver Aspekt des *IBM Workplace* Komponentenmodells ist die weitreichende Unterstützung von Standards. Hahnl sieht in der häufig fehlenden Benutzung von Standards portalbasierter Systeme das Problem der mangelnden Integrationsfähigkeit von Kollaborationsapplikationen. Gerade spezielle Produkte von Drittanbietern können ohne eingehaltene Standards nur schwer integriert werden. [vgl. Hahnl 2004 S.91]

4.2.2.2 Objektbasierte Integration

Die Integration kollaborativer Funktionalität auf Objektebene, d.h. eine Integration in das Kontextmenü bestimmter Objekttypen, wird auch in *IBM Workplace* genutzt um die

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

verschiedenen Komponenten und Applikationen mit kollaborativen Funktionalitäten anzureichern.

Diese Form der Integration wird in *IBM Workplace* insbesondere für die Objekte vom Typ „Person“ konsequent genutzt. Das dynamische Kontextmenü wird in allen Browser-basierten *Workplace* UIs durch die Verwendung des so genannten „Person-Tag“ realisiert. Das Person-Tag ist ein JSP-Tag welches einen damit umgebenen Personennamen zu einem so genannten „Live Name“ macht und ihn mit einem klicksensitiven Menüicon versieht. Wird dieses Icon angewählt, so erscheint ein Kontextmenü, welches die verschiedensten, auf diese Person anwendbaren Funktionen bereitstellt. Dazu gehören neben Funktionen zum Anzeigen detaillierter Informationen auch Funktionen um eine E-Mail an diese Person zu senden oder einen Chat zu starten. Dabei ist das Person-Tag Kontext-aware, d.h. es werden nur die Funktionen im Menü angeboten, die aufgrund des Online-Status der Person nutzbar sind.

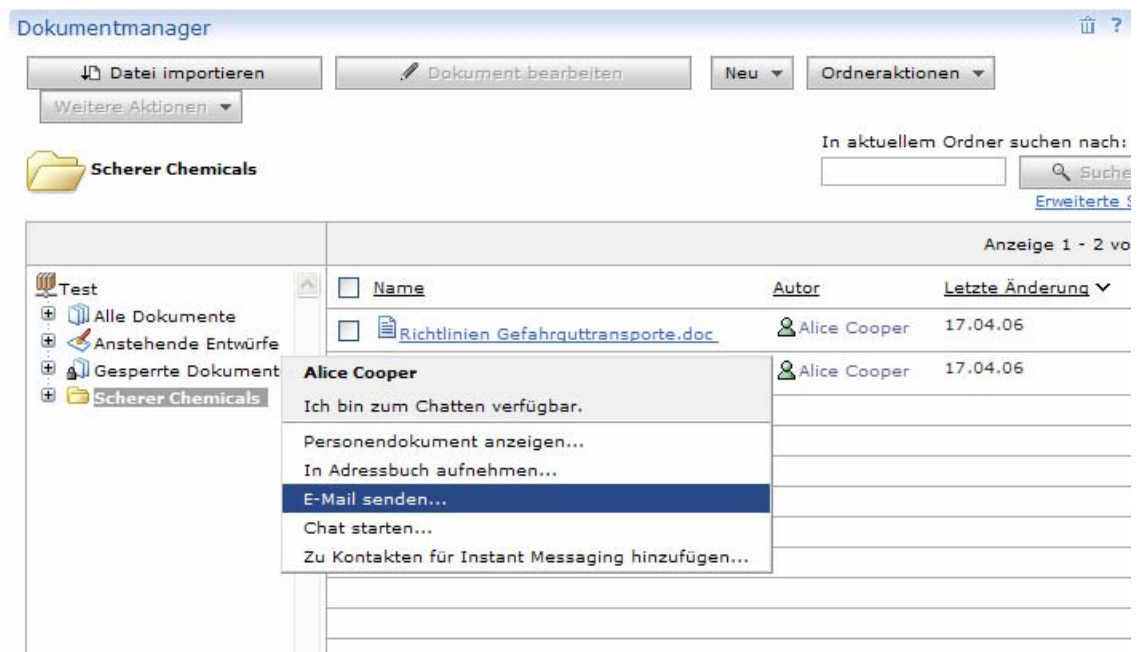


Abb. 23: Kollaborative Funktion integriert in das Kontextmenü einer Person

Ist eine Person z.B. nicht online, so wird die Funktion zum starten eines Chat nicht angeboten. Für die Eclipse basierten UIs im *Workplace Managed Client* existiert ein entsprechendes Eclipse-Control mit dem Namen „LiveNameViewer“ welches die gleiche Funktionalität wie das Person-Tag bereitstellt.

Die Integration kollaborativer Funktionalität auf Objektebene wird für den Typ „Person“ konsequent von den verschiedenen *IBM Workplace* Applikationen und Komponenten-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

ten genutzt, sowohl in den Browser basierten UIs als auch im *Workplace Managed Client*. Dagegen werden Objekte anderen Typs, wie z.B. Dokumente, Diskussionsbeiträge, E-Mails oder Teamspaces im Browser UI gar nicht und im *Workplace Managed Client* nur teilweise mit Kontextmenüs versehen, die eine Integration von kollaborativen Funktionen erlauben. So haben im *Workplace Client* etwa E-Mail Objekte die Funktion „Chat“ in das Kontextmenü integriert, durch das der Anwender einen Chat mit dem Absender der E-Mail starten kann. Ebenso ist eine Chat-Funktion im Kontextmenü eines Dokumentes in einer Dokumentenbibliothek verfügbar um dessen Autor eine Instant Message zu schicken.

Bewertung

Allerdings wird deutlich, dass für diese Objekte kein einheitliches Tag verwendet wurde. So stehen im Kontextmenü eines Dokumentes im Activity-Explorer teilweise andere kollaborativen Funktionen zur Verfügung als im Kontextmenü eines Dokumentes in einer Dokumentenbibliothek. Außerdem werden nicht alle verfügbaren kollaborativen Komponenten, die *IBM Workplace* bietet, in den Kontextmenüs der Objekte angeboten. Es zeigt sich dadurch, dass die Integration auf Objektebene hier von jeder Komponente anders implementiert wird, und im Gegensatz zu den Live-Names für Personen hier kein generisches „Live-Objekt“-Tag Verwendung findet.

Dies zeigt sich auch in der Verfügbarkeit entsprechender Mechanismen im *Workplace SDK 2.6*. Während die Live-Name Funktionalität für Personen durch ein einfach zu benutzendes JSP Tag bzw. ein Eclipse-Control realisiert ist und im SDK auch für andere Entwickler zur Verfügung steht, existieren keine entsprechenden Tags für Objekttypen wie z.B. Dokumente oder andere Informationsobjekte.

Die Live-Name Technologie kann sehr einfach in eigene *Workplace* Komponenten integriert und von dort genutzt werden. Allerdings erlaubt auch das Person-Tag keine Erweiterung des Kontextmenüs durch eigene kollaborative Komponenten. Sie sind daher auf die im Lieferumfang von *IBM Workplace* mitgelieferten kollaborativen Funktionalitäten beschränkt.

Damit entsprechen die in *IBM Workplace* bereitgestellten Mechanismen zur Integration von kollaborativen Funktionen auf Objektebene nur teilweise den im idealtypischen Framework definierten Anforderungen der Erweiterbarkeit durch eigene kollaborative Funktionen sowie der Verfügbarkeit für jegliche Objekttypen.

4.2.3 Interkomponentenkommunikation

Wie in Kapitel 4.2.1 erläutert werden Komponenten in *IBM Workplace* als Portlets oder als Plugins implementiert und integriert. Dies ergibt sich aus der Portal-Architektur. Kapitel 4.2.2 beschreibt die verschiedenen von *IBM Workplace* angebotenen Möglichkeiten die Komponenten in ein bestehendes System zu integrieren um neue Funktionalitäten bereitzustellen. Oft ist es neben der Integration aber zusätzlich auch sinnvoll, dass Komponenten gegenseitig interaktiv handeln und Daten austauschen können. Dabei ist im Rahmen der kontextuellen Kollaboration besonders der Austausch von Kontextinformationen von Bedeutung. Ein Beispiel hierfür ist der Austausch von Daten zwischen einer CRM-Komponente, die eine Liste von Kunden anzeigen kann und einer Mail-Komponente. Klickt der Anwender in der CRM-Komponente auf einen Kundennamen, so kann sich die Inbox in der Mailkomponente auf diese Auswahl kontextualisieren und nur E-Mails anzeigen, die von diesem Kunden stammen. Diese Interaktion ist aber nur möglich, wenn die beiden Komponenten miteinander kommunizieren können.

4.2.3.1 Interportletkommunikation

Workplace stellt verschiedene Möglichkeiten der Interportletkommunikation zur Verfügung: Die API-basierte Definition der Kommunikation und das Konzept der kooperativen Portlets.

Bei der API basierten Interportletkommunikation handelt es sich um eine programmierseitig hart verdrahtete Definition der Kommunikation. Diese Variante bedeutet, dass die Portlets zusammen auf einer Portalseite liegen müssen und dass die Portlets nicht ohne weiteres von einander getrennt werden können. Eine nachträgliche Definition von Interportletkommunikation ist mit dieser Methode kaum möglich. Die Verdrahtung muss in der Regel von einem Programmierer vorgenommen werden. Die Vor- und Nachteile dieser Variante gehen mit denen der Integration auf API-Ebene einher [vgl. Kapitel 4.2.2.1] und sollen daher hier nicht näher erläutert werden.

Das Konzept der kooperativen Portlets ist in *IBM WebSphere Portal* realisiert und wird daher auch in *IBM Workplace* unterstützt. Kooperative Portlets unterstützen eine Möglichkeit, Daten zwischen zwei oder mehreren Portlets auszutauschen. Anders als bei Interportletkommunikationsarten, die programmierseitig fest erstellt werden, wie beim Messaging, sind die Bindungen der kooperativen Portlets weniger stark. Zwar werden zur Entwicklungszeit die grundsätzlichen Austauschfähigkeiten festgelegt, aber es wer-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

den keine expliziten Verbindungen erstellt. D.h. es muss nicht zur Programmierzeit feststehen, mit welchen Portlets das gerade entwickelte Portlet kommunizieren soll. Koooperative Portlets beschreiben ein Modell zum Deklarieren, Veröffentlichen und gegenseitigem Austausch von Informationen. Dadurch ist es möglich, neue Portlets zur Laufzeit eines Systems in die Interportletkommunikation einzubinden und dann festzulegen, welche Portlets miteinander kommunizieren sollen.

Das grundsätzliche Prinzip der Kommunikation basiert auf der in Kapitel 3.3.4 genannten „coordinated interaction“. Sie verläuft so, dass es möglich ist, in den Portlets bestimmte Daten als **Properties** zu definieren. Properties sind Teil des Datenmodells der Portlets. D.h. diese Daten kann ein Portlet empfangen, versenden oder beides. Z.B. wäre in dem obigen Beispiel des CRM-Portlets und des Mail-Portlets der Kundename als ein solches Property definiert. Wichtig ist, dass diese Properties zwar bereits zur Programmierzeit als Property deklariert werden, aber erst in der Benutzungsphase kann bestimmt werden, welche Werte die Properties erhalten.

Außerdem werden für die jeweiligen Properties **actions** definiert. Das sind die Aktionen, die ausgeführt werden, wenn das dazugehörige Property empfangen oder versendet wurde. In dem Beispiel des Mail-Portlets ist die Aktion so definiert, dass das E-Mail-Portlet beim Empfang des Properties, in diesem Fall der Kundename, dieses als Anzeige-Filter nutzt. Der Empfang eines Properties kann eine oder mehrere Aktionen auslösen oder eine Action kann ein oder mehrere Properties erzeugen, oder beides.

Actions sind verbunden mit **Parametern**. Die Parameter bestimmen, welche Aktion auf welchem Property ausgeführt wird und sie definieren auch, auf welche Art ein Property von einer Action genutzt oder erzeugt wird. Bezogen auf das E-Mail-Portlet Beispiel bedeutet dies, dass auf dem Property „Name“ die Aktion „Name als Anzeigefilter verwenden“ in dem E-Mail-Portlet definiert ist. Auf dem Property „Adresse“ könnte z.B. in einem Routenplaner-Portlet die Aktion definiert sein „Zeige die Route zu der Adresse“.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

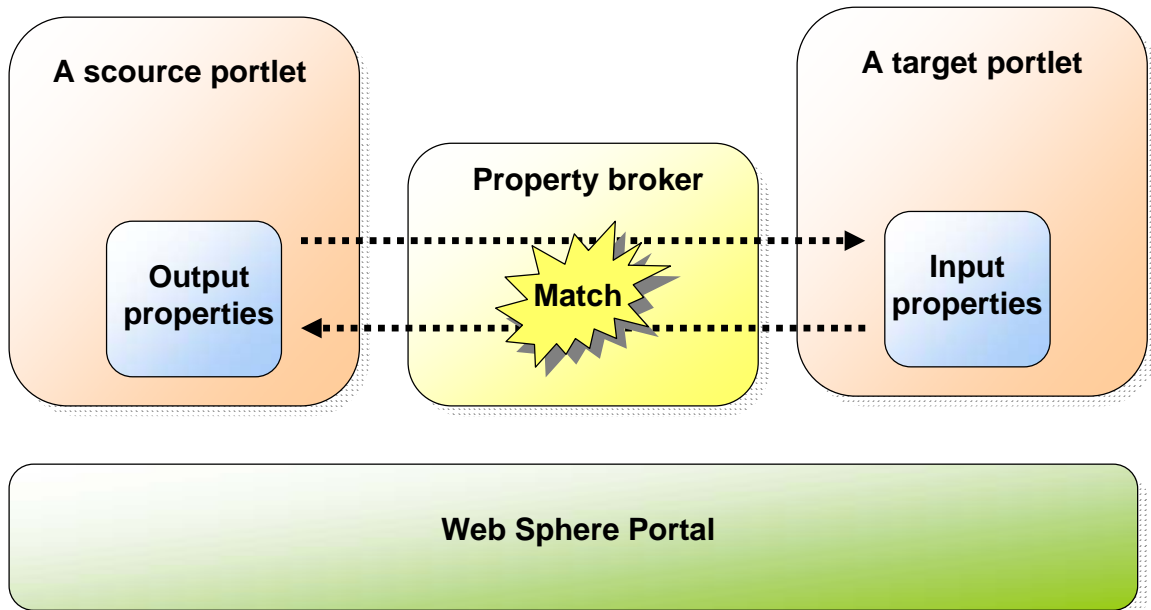


Abb. 24: Interportletkommunikation mit Hilfe des Property Broker

Wichtig für die kontextuelle Kollaboration ist an dem Konzept der kooperativen Portlets dass diese Properties nicht direkt von Portlet zu Portlet übergeben werden, sondern es steht der Property-Broker, ein Portlet-Service, zur Verfügung, der den entkoppelten Austausch von Properties zwischen verschiedenen Portlets ermöglicht. Das bedeutet, dass für die Portlets nicht relevant ist, mit welchen anderen Portlets sie Daten austauschen. Jedes Portlet registriert sich nur bei dem Property-Broker durch die Veröffentlichung seiner Properties, die es empfangen oder senden kann. Der Property-Broker regelt dann die Weiterleitung der Daten an andere Portlets, die diese Daten empfangen können. Abb. 24 zeigt den grundsätzlichen Mechanismus. Durch dieses Verfahren wird die Flexibilität der Integration deutlich verbessert. Das bedeutet für ein kontextuelles Kollaborationssystem, dass es möglich wird spezielle Komponenten einer Applikation hinzuzufügen und eine Kommunikation zu definieren, sobald ein abzuwickelnden Geschäftsprozess dies verlangt. Die Applikation kann sich dadurch auf diesen Prozess kontextualisieren. Einzige Voraussetzung dafür ist die grundsätzliche Implementation von Properties und Actions. Darüber hinaus ist auch das Entfernen von nicht genutzten Portlets unkompliziert möglich, da die anderen Portlets auch nach der Entfernung problemlos weiter arbeiten.

Das kooperative Verhalten von Portlets wird eher durch Konfiguration integriert als durch Programmierung. Die Konfiguration wird dabei nicht von dem Programmierer vorgenommen, sondern kann von einem Administrator oder auch dem Anwender selbst

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

durchgeführt werden. *IBM WebSphere Portal* stellt dafür das Wiring-Tool-Portlet zur Verfügung, welches eine einfache Definition der Kommunikation möglich macht. Dadurch liegt die Integration und Definition von Interportletkommunikation in der Hand des Anwenders. Dieser besitzt das Kontextwissen über die Business-Prozesse und weiß damit welche Portlets auf welche Art miteinander verbunden sein sollten, damit sie sich auf optimale Weise auf den Business-Prozess kontextualisieren können. [vgl. zu diesem Kapitel: IBM InfoCenter (1) 2006]

Bewertung

Bei der Evaluation der in *IBM Workplace* realisierten Interportletkommunikation zeigt sich ein uneinheitliches Bild.

Das vorgestellte Konzept der kooperativen Portlets und die Bereitstellung des *Property-Broker* und dem *Wiring-Tool* bieten die in Kapitel 3.3.3 und Kapitel 4.2.3.1 hinreichend erläuterten Vorteile. Die Interportletkommunikation kann somit wie beschrieben auf mehrere Arten in einem auf *IBM Workplace* basierenden System definiert und verwendet werden.

Allerdings ist dieses Konzept in *Workplace* nicht konsequent durchgeführt, da die meisten der mitgelieferten *Workplace* Komponenten die Interportletkommunikation für sich selbst noch nicht nutzten d.h. diese Komponenten haben keine Properties definiert. Daher lässt sich eine Kommunikation mit diesen Portlets nicht nachträglich implementieren. Die flexible und zur Laufzeit definierbare Interportletkommunikation ist zwar und für selbst zu entwickelnde Portlets nutzbar, sobald sie *Properties* und *Actions* definiert haben, allerdings lassen sich auf diese Weise bisher keine *IBM Workplace* Portlets mit selbst entwickelten Komponenten kooperativ verbinden. *IBM Workplace* nutzt an dieser Stelle das Potential nicht aus, um kontextuelle Kollaboration zu unterstützen. [vgl. IBM d 2006]

4.2.3.2 Interplugin-Kommunikation

Der *IBM Workplace Managed Client* implementiert ein anderes Komponentenmodell als die serverseitigen Applikationen der portalbasierten *Workplace Collaboration Services*. Da der *Workplace Managed Client* auf der Eclipse Technologie basiert, nutzt er auch das dort gängige Komponentenmodell. In Eclipse werden alle Komponenten als Eclipse Plugins implementiert. Die Eclipse Technologie wurde explizit mit dem Ziel der

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Erweiterbarkeit entworfen. [vgl. Gamma/Beck 2004, S. 35] Die Erweiterbarkeit wird durch so genannte Erweiterungspunkte (Extension-Points) und Erweiterungen (Extension) erreicht. Extension-Points definieren dabei eine Schnittstelle, an der sich das System erweitern lässt, während Extensions jeweils eine spezifische Implementation eines solchen Extension-Points darstellen. Damit stellen Extension-Points im Grunde eine SPI dar. Jedes Plugin, also auch jede Komponente im *Workplace Managed Client*, kann verschiedene Extensions implementieren. Sowohl Eclipse als auch der *Workplace Managed Client* selber definieren eine ganze Reihe solcher Extension-Points für die verschiedensten Zwecke.

Um eine Plugin-zu-Plugin Kommunikation zu realisieren, können Plugins selber den von Eclipse zur Verfügung gestellten Erweiterungsmechanismus nutzen und eigene Extension-Points definieren die dann von anderen Plugins implementiert werden können. Diese Extension-Points legen das Interface fest durch welches das eine Plugin mit beliebigen anderen Plugins kommunizieren kann, welche eine passende Erweiterung zu diesem Extension Point implementieren. Die Kommunikation zwischen den Plugin-Komponenten basiert dann darauf, dass die eine Komponente die durch den Extension - Point definierte API der anderen Komponente aufruft.

Bewertung

Das hier beschriebene Verfahren der Interplugin-Kommunikation basiert auf der in Kapitel 3.3.3.1 beschriebenen Integration auf API Ebene. Während dieses Verfahren in Eclipse sehr flexibel ist und durch die Definition von Extension-Points eine gewisse Entkopplung der beiden Komponenten voneinander ermöglicht, so beinhaltet sie doch auch alle Nachteile der API basierten Integration: Die Komponenten müssen, wie bei der Nutzung einer SPI, zur Implementierungszeit zumindest den Extension Point kennen über den sie kommunizieren. Dadurch können später zwar Komponenten ausgetauscht werden, die den gleichen Extension point implementieren, aber es können keine Komponenten miteinander kommunizieren die neue oder unterschiedliche Extension-Points nutzen. Es existiert in Eclipse und im *Workplace Managed Client* keine vermittelte (brokering) Kommunikation, wie es der Property Broker in WebSphere Portal ermöglicht. Dadurch ist es z.B. nicht möglich, dass ein Anwender mehrere unabhängige und sich unbekannte Plugins ohne zu programmieren durch das Erzeugen einer Verbindung („Wire“) miteinander verbinden und diese so Kontextinformationen austauschen könnten.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Der *Workplace Managed Client* definiert allerdings eine Reihe generischer Extension-Points die es Komponenten erlauben sich in das User Interface des Clients zu integrieren. Dazu gehören z.B. Extension-Points um neue Menüs und Menüeinträge zu erstellen oder neue Actions-Buttons für die Action-Bar des *Workplace Client* zu definieren.

4.2.4 Gemeinsam genutzte Plätze - Shared-Spaces

In *IBM Workplace* bilden Shared-Spaces, neben den Unterstützungsfunktionen für die Realtime-Kollaboration, das wichtigste Element der kontextuellen Kollaboration. *IBM Workplace* stellt verschiedene Arten von Shared-Spaces zur Verfügung, die als Kontext für die Bearbeitung verschiedener Geschäftsprozesse dienen können. Entsprechend der Systematik in Kapitel 3.3.5.2 *Asynchrones Sharing durch Shared-Spaces* sind dies:

Ad-hoc orientierte, activity-centric Spaces

Ad-hoc orientierte Shared-Spaces werden in *IBM Workplace* durch Activities realisiert. Activities werden im Activity-Explorer bearbeitet, einem *Workplace Managed Client* basierten Kollaborationswerkzeug. Activities erlauben es einem oder mehreren Personen verschiedene zu einem Geschäftsprozess gehörende Dokumente oder andere Objekte in einer Activity zusammenzufassen und sie mit den anderen Mitgliedern der Activity, sowohl asynchron als auch synchron, gemeinsam zu bearbeiten. Activities unterstützen fünf Typen von Objekten die in einer Activity angelegt und bearbeitet werden können.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

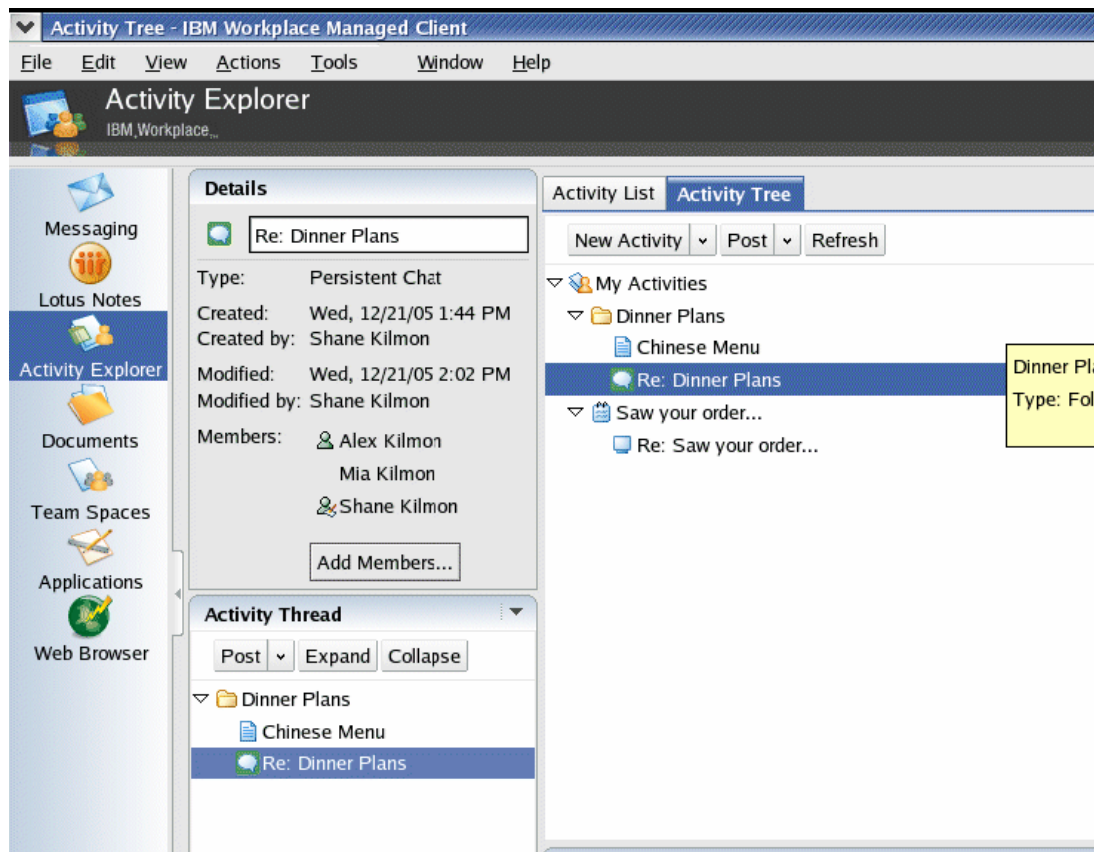


Abb. 25: IBM Workplace Activity Explorer

[IBM b 2006]

Dazu gehören:

- Persistente Chats
- Dateien (Files)
- Verzeichnisse (Folder)
- Anmerkungen (Notes)
- und Shared-Screenshots

Wird eine neue Activity angelegt so ist sie zunächst vollkommen unstrukturiert. Die Mitglieder können dann die oben genannten Objekte anlegen und gemeinsam asynchron nutzen. Sie können Verzeichnisse und Unterverzeichnisse anlegen um die verschiedenen Objekte in der Activity zu strukturieren. Außerdem können Objekte als Antwort auf bereits bestehende Objekte angelegt werden, wodurch eine einem Diskussionsforum ähnliche Hierarchie von gemeinsam genutzten Objekten in der Activity entsteht. Beim Anlegen eines neuen Objektes kann der Anwender eine Liste von Mitgliedern definieren, mit denen er das Objekt gemeinsam nutzen möchte. Das Konzept der Mitglied-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

schaft lässt sich in Activities also für jedes einzelne Objekt definieren, aber auch auf ganze Teilbäume der Objekthierarchie oder die gesamte Activity beziehen.

Activities nutzen viele der in Kapitel 3.2 genannten Strukturmerkmale für die kontextuelle Kollaboration.

Abhängig vom Objekttyp können die Mitglieder die Objekte auch synchron bearbeiten. Persistente Chats und Shared-Screenshots können gleichzeitig bearbeitet werden, wenn mehrere Mitglieder das gleiche Objekt zur gleichen Zeit öffnen.

Es stehen für alle Personen in einer Activity Presence-Awareness Funktionalitäten zur Verfügung. Die gemeinsam genutzten Objekte in der Activity sind mit Resource-Awareness versehen. So zeigen sie einem Anwender z.B. an wenn ein anderes Mitglied ein Objekt gerade bearbeitet oder wenn es verändert wurde. Über diese Änderungen kann sich der Anwender auch dann per Notification benachrichtigen lassen wenn er sich selbst gerade in einem anderen Kontext befindet.

Activities lassen sich im *IBM Workplace Managed Client* sehr einfach erstellen. So kann man z.B. durch das Ziehen eines Dokumentes aus einer Dokumentenbibliothek auf das Icon einer Person eine Activity erzeugen in der das Dokument mit der ausgewählten Person gemeinsam genutzt wird.

Somit sind Activities sehr gut für spontane und unstrukturierte Geschäftsprozesse geeignet die sich durch die vielfältigen synchronen und asynchronen Kollaborationsmechanismen sehr kurzfristig und effizient im Team bearbeiten lassen. Sie sind ebenfalls gut geeignet für Vorgänge, die sich erst während der normalen Kollaboration zwischen 2 Personen ergeben. Wenn z.B. ein Anwender einem Kollegen ein Dokument per E-Mail zusendet mit der Bitte um Kommentierung und dieser aber einen dritten Kollegen mit heranziehen möchte so ist dies eine geeignete Gelegenheit aus dem Dokument eine Activity zu starten und die drei Personen als Mitglieder hinzuzufügen.

Semi-strukturierte Shared Workspaces

Zur Kollaboration in häufiger auftretenden Geschäftsprozessen oder längerfristigen Projekten bietet *IBM Workplace* die *Workplace* Teamspaces. Sie basieren genau wie die *Workplace* Applikationen auf dem Prinzip der Composite-Applications. Im Gegensatz zu Activities lassen sich Teamspaces wie in Kapitel 4.2.2 beschrieben auf Basis von vordefinierten oder selbst erstellten Templates anlegen und durch das Hinzufügen zu-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

sätzlicher Komponenten an den gegebenen Geschäftsprozess anpassen. Diese Komponenten können sowohl kollaborative Funktionen bereitstellen als auch komponentisierte Business-Applikationen darstellen. Dadurch können den Mitgliedern eines Teamspaces genau die Werkzeuge und kollaborativen Komponenten bereitgestellt werden, die sie zur Bearbeitung des jeweiligen Geschäftsprozesses benötigen.

IBM Workplace liefert bereits eine Reihe kollaborativer Komponenten als Teil von *IBM Workplace* mit. Diese unterstützen zumeist die asynchrone Kollaboration, wie z.B. eine Diskussionsforumskomponente oder eine Dokumentenbibliothekskomponente. *Workplace* liefert aber auch Komponenten zur asynchronen Kollaboration als Teil eines Teamspaces. Dazu gehören z.B. eine Gruppen-Chat Komponente oder eine Web-Konferenz Komponente. Dabei werden bereits eine größere Anzahl vordefinierter Templates bereitgestellt, auf deren Basis neue Teamspaces erstellt werden können. Der Anwender hat zudem die Möglichkeit eine Liste von Mitgliedern für einen Teamspace zu definieren und diesen Mitgliedern Rollen zuzuweisen.

Formale, Workflow getriebene Shared-Spaces

IBM Workplace bietet derzeit keine Funktionalität um einen formalen Geschäftsprozess durch einen Workflow zu unterstützen. Daher eignet sich *IBM Workplace* vor allem zur kontextuellen Kollaboration in unstrukturierten bis semi-strukturierten Prozessen, für die Koordination und Unterstützung strukturierter, formaler Geschäftsprozesse muss allerdings auf Komponenten von Drittherstellern zugegriffen werden.

Bewertung

Die dritte Stufe der kontextuellen Kollaboration zeichnet sich unter anderem durch die Persistierung von Kontextbeziehungen aus. Shared-Spaces wurden in Kapitel 3.3.5.2 als geeignetes Werkzeug identifiziert, um diese Kontextbeziehungen zu unterschiedlichen Objekten zu erstellen und zu speichern.

Dabei wurden drei Persistierungsarten, entsprechend den unterschiedlichen Arten von Kontextbeziehungen identifiziert (vgl. 3.2.3):

- Persistierung von Kontextbeziehungen zu Dokumenten, die im Kontext des Prozesses entstehen
- Referenzieren von Kontextbeziehungen zu Objekten, die zu einem anderen Kontext gehören

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

- Hinzufügen von Kontextbeziehungen, zu Objekten die keinem Geschäftsprozess angehören

Die Evaluation der Realisation von Shared-Spaces in der *IBM Workplace* Architektur zeigt eine teilweise Erfüllung der an dieses Strukturmerkmal gestellten Anforderungen. In Kapitel 3.3.5.2 wurde dargestellt, dass die unterschiedlichen Formen von Geschäftsprozessen ein Kontinuum von unstrukturierten zu formalen Prozessen aufspannen. Um diese verschiedenen Prozessformen optimal abzubilden, wurde erläutert, dass ein kontextuell-kollaboratives System nach Möglichkeit

- Ad-hoc orientierte, activity-centric Spaces,
- semi-strukturierte Shared-Workspaces ,
- und formale, workflow-getriebene Spaces

bereitstellen sollte.

Die **Ad-hoc orientierten, activity-centric Spaces** werden, wie vorangegangen erläutert wurde, in der *IBM Workplace* Architektur durch den Activity-Explorer repräsentiert. Die Ausführungen machen deutlich, dass es mit dem Activity-Explorer grundsätzlich möglich ist, ad hoc einen neuen Geschäftsprozess zu initiieren und die dafür benötigten Dokumente in diesen Space für die gemeinsame Nutzung einzustellen. Der Geschäftsprozess muss dabei keine Struktur mitbringen. Die Definition von Mitgliedschaften (Membership) zu einer Aktivität ist möglich. Damit erfüllt der Activity-Explorer einen Teil der an ihn gestellten Anforderungen.

Die unterschiedlichen Arten der Persistierung können im Activity-Explorer nur teilweise realisiert werden. Da der Activity-Explorer derzeit nur genau fünf unterschiedliche Objekttypen definieren kann, ist es nicht möglich, Objekte anderen Typs in eine Aktivität einzubinden.

Auch Objekte vom Typ „Referenz“ lassen sich derzeit nicht abspeichern. Das bedeutet, dass auch keine Kontextbeziehungen zu Objekten herstellbar sind, die zu einem anderen Geschäftsprozess gehören. Das kann nur durch eine Kopie dieser Objekte erfolgen.

Referenzen auf externe Objekte oder auf Objekte in externen Systemen können im dem Activity-Explorer nicht hinzugefügt werden. So lässt sich auch eine Aktivität nicht in eine andere Aktivität einbinden.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Damit wird die aufgestellte Forderung nach Abspeicherung von Referenzen nicht erfüllt.

Nachteilig im Sinne der kontextuellen Kollaboration ist darüber hinaus zu bewerten, dass der Activity-Explorer (Version 2.6) nur Desktop-basiert im *IBM Workplace Managed Client* zur Verfügung steht. Dies hat den Nachteil, dass Mitarbeiter, die vollständig Browser-basiert arbeiten, ausgeschlossen werden und daher nicht in Activities eingebunden werden können.

Semi-strukturierte Shared Workspaces werden wie beschrieben durch die *IBM Workplace* Teamspace realisiert. Sie stehen sowohl Browser-basiert als auch im *IBM Workplace Managed Client* zur Verfügung.

Die Evaluation hat gezeigt, dass die *IBM Workplace* Teamspace dazu geeignet sind, semi-strukturierte Geschäftsprozesse zu unterstützen und die in ihnen entstehenden Kontextbeziehungen zu persistieren. Dabei dient das auch in den Teamspace angewendete Konzept der „Composite-Application“ [vgl. Kapitel 4.2.2.2] dazu, die geforderte Strukturierung vorzunehmen.

Anders als bei dem Activity-Explorer, sind Teamspace durch Komponenten erweiterbar. Daher ist es hier möglich neue Komponenten hinzuzufügen, und so eine Vielzahl von Objekttypen gemeinsam nutzen zu können.

Die Evaluation hat weiter gezeigt, dass die unterschiedlichen Formen der Persistierung von Kontextbeziehungen, genau wie im Activity-Explorer, bisher nur teilweise vorgenommen werden können. So ist es derzeit in den mitgelieferten Komponenten nicht möglich, die als Anforderung genannten Referenzierungen auf externe Objekte vorzunehmen. Die Teamspace erfüllen daher nur teilweise die an sie gestellten Anforderungen.

Da in der aktuellen Version von *IBM Workplace Collaboration Services* keine **Spaces für formale, Workflow getriebene Geschäftsprozesse** vorgesehen sind, können die Beziehungen zwischen Kontextinformationen und einem formalen Prozess auf diese Art nicht gemanaged werden. Diese Anforderung an ein kontextuell-kollaboratives System wird von der *IBM Workplace* Plattform derzeit nicht erfüllt.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

4.2.5 Membership

Das Konzept der Mitgliedschaft findet sich in *IBM Workplace* durchgängig für alle Arten von Shared-Spaces. Dazu gehören die *Workplace* Applikationen, Teamspaces und Web-Konferenzen sowie Activities und Dokumentenbibliotheken. Die Liste der Mitglieder wird zunächst von der Person festgelegt die den jeweiligen Shared-Space angelegt hat. Jede Person die dabei als Mitglied dem Shared-Space hinzugefügt wird, wird dabei einer Rolle zugewiesen, die sie in diesem Space besitzt. Diese Rollen sind typischerweise vorgegeben und hängen vom Typ des Shared-Space ab. So stellen Teamspaces die Rollen „Moderatoren“ und „Mitwirkende“ zur Verfügung während zum Beispiel die Web-Konferenzen die Rollen „Vorsitzender“, „Moderatoren“ und Teilnehmer“ kennt. Einzig Activities unterstützen kein Rollenkonzept für die Mitglieder. Die Mitglieder werden durch das Mitglieder Portlet dem Space hinzugefügt und einer Rolle zugewiesen.

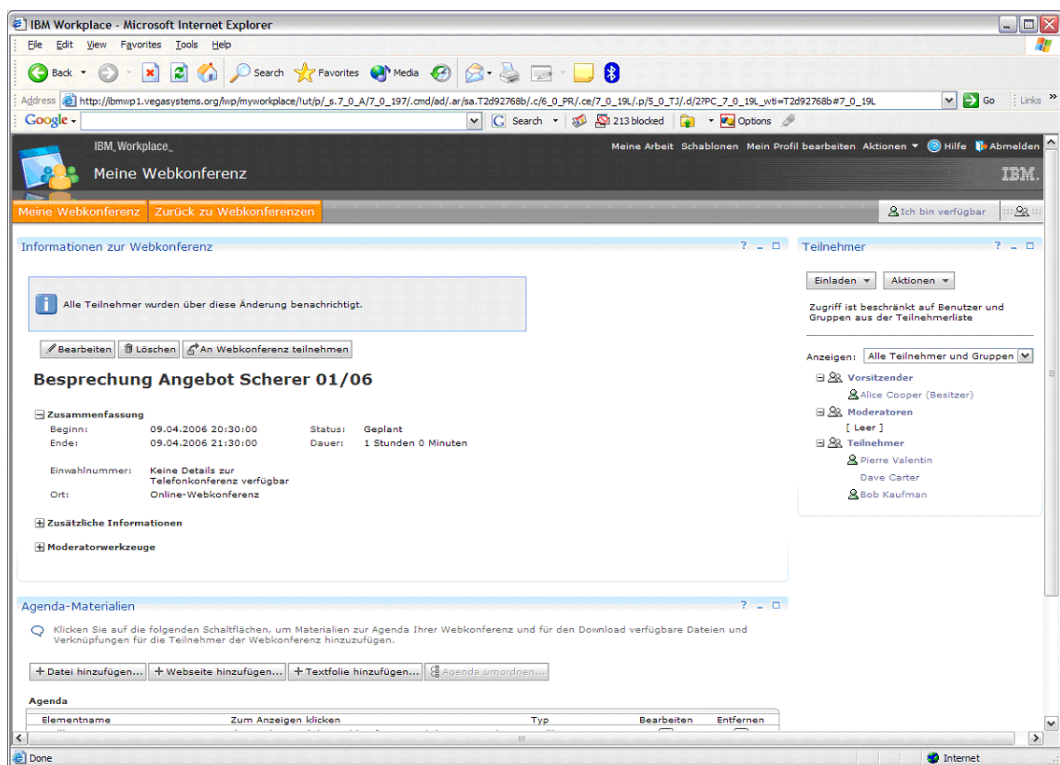


Abb. 26: Membership-Portlet mit Rollen im Teamspace

Während die Rollen für die meisten Shared-Space Typen festgelegt sind lassen sie sich für *Workplace* Applikationen und Teamspaces anpassen, erweitern und selber definieren. An jede Rolle sind bestimmte Zugriffsrechte gebunden mit der jede Person, die

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

diese Rolle besitzt, ausgestattet wird. Diese Zugriffsrechte beziehen sich auf die Sichtbarkeit von einzelnen Portlets und ganzen Seiten in der Applikation bis hin zu ganz individuellen Funktionen und Aktionen innerhalb der einzelnen Komponenten, die einigen Rollen zur Verfügung stehen, anderen aber nicht. Daher führt die Nutzung von Rollen in den Teamspaces zu ganz individuellen, rollenabhängigen Benutzererfahrungen, die die Mitglieder in einem Shared-Space haben. So sieht ein Moderator eines Teamspaces zum Beispiel die Seite „Mitgliederverwaltung“, die eine Person mit der Rolle „Mitwirkender“ nicht sieht, oder er hat in einem Diskussionsforum die Möglichkeit, Beiträge und ganze Foren zu löschen, was ebenfalls nicht von einem Mitwirkenden durchgeführt werden kann.

Welche Rechte und Fähigkeiten eine Rolle auf einer bestimmten Komponente in *Workplace* hat, kann von der Implementation der Komponente selber festgelegt werden. Dazu muss die Komponente das Interface „Membership“ implementieren, welche eine der *Workplace* Business-Component-Interfaces ist. . [Groger/Gorman 2004 b, S.5] Durch dieses Interface definiert der Programmierer der Komponente eine Menge von Komponentenrollen, d.h. speziellen Rollen, die lokal für die Komponente Sinn machen, und kann diesen Rollen innerhalb der Implementation der Komponente bestimmte Rechte zuweisen. So kann eine Projektmanagement-Komponente z.B. die Komponentenrollen „Manager“, „Autor“ und „Leser“ bereitstellen. Wird diese Komponente später mit Hilfe des *Workplace Builder* Werkzeugs zu einem Teamspace- oder Applikationstemplate hinzugefügt, so kann der Benutzer, der dieses Template erstellt die Komponentenrollen an Applikationsrollen zuweisen. Applikationsrollen sind vom Templateersteller definierte logische Rollen, die für die gesamte Applikation Sinn machen. Jeder Komponentenrolle wird dann eine Applikationsrolle zugewiesen, wobei mehrere Rollen verschiedener Komponenten an eine Applikationsrolle zugewiesen werden können. So dienen die Applikationsrollen zur logischen Aggregation der verschiedensten Komponentenerollen. Der Benutzer, der schließlich aus einem solchen Template eine Applikationsinstanz erstellt, kann dann den einzelnen Mitglieder die Rollen zuweisen, und ihnen so indirekt die verschiedenen Rechte auf die Komponenten geben.

Zur Laufzeit können die verschiedenen Komponenten innerhalb einer solchen Composite-Application durch diesen Mechanismus auf die Rolle des aktuellen Benutzers kontextualisieren, der gerade mit der Applikation arbeitet. So sind alle Komponenten automa-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

tisch in der Lage dem Benutzer nur die Fähigkeiten und Funktionen anzubieten, die in dem aktuellen Kontext für den Benutzer relevant bzw. erlaubt sind.

Neben der Kontextualisierung der Komponente auf die Rolle und damit die Rechte eines Benutzers in einem bestimmten Kontext hat die Mitgliedschaft noch eine weitere Bedeutung für die kontextuelle Kollaboration in *IBM Workplace*. Die Mitgliedschaft an einem Shared-Space drückt neben den Aspekten einer Access-Control-List auch das Interesse der Mitglieder an dem gegebenen Kontext aus. Mitglieder eines Kontextes sind damit typischerweise die Personen die auch aktiv in die Bearbeitung des Kontext involviert sind und Wissen über diesen Business Prozess haben. Damit sind sie typischerweise auch genau die Gruppe von Personen, die als Kommunikations- und Kollaborationspartner im Kontext dieses Business Prozess von Interesse sind, z.B. um ein Problem bezüglich des Business Prozesses in einem Chat zu klären oder eine Rückfrage zu dem Vorgang per E-Mail an einen der Prozessbearbeiter zu stellen.

Bewertung

Die Evaluation hat gezeigt, dass das Konzept der Membership sehr durchgängig in allen Komponenten angeboten wird. Damit ist dieses Strukturmerkmal eines kontextuell-kollaborativen Frameworks in der *IBM Workplace* Architektur hinreichend realisiert.

4.2.6 Kontext-Awareness

Wie in Kapitel 3.3.7 beschrieben gliedert sich die Kontext-Awareness in die Bereiche People-Awareness sowie Ressource-Awareness auf. *IBM Workplace* verwendet den Aspekt der People-Awareness - und hier speziell die Online-Awareness - sehr konsequent und durchgehend im gesamten *Workplace* Produktportfolio.

4.2.6.1 Online-Awareness

Die Online-Awareness wird in *IBM Workplace* ebenfalls durch das „Person-Tag“ realisiert, welches auch das Kontextmenü für die Person bereitstellt (vgl. Kapitel 3.3.3.3). Das Person-Tag versieht, genau wie das entsprechende Live-Name-Control im *Workplace Managed Client*, einen damit umgebenen Personennamen zusätzlich mit einem Awareness-Icon. Dieses Awareness-Icon zeigt in Echtzeit den Online-Status der entsprechenden Person – also ob sie im *Workplace* System angemeldet ist oder nicht, und, falls sie angemeldet ist, ob sie gerade verfügbar ist, gerade abwesend von ihrem

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Arbeitsplatz ist oder ob sie nicht gestört werden möchte. Diese Onlinezustände werden durch verschiedenfarbige Awareness-Icons repräsentiert.

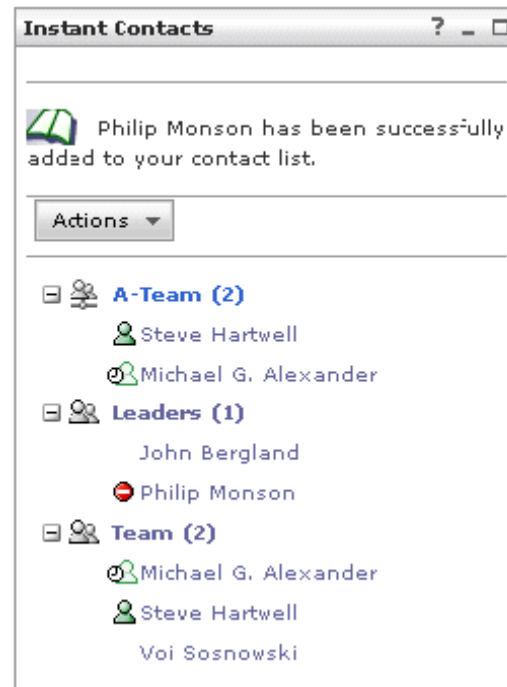


Abb. 27: Instant Contacts mit Online-Awareness-Funktionalität
[IBM c 2006]

Die Awareness-Funktion unterstützt dabei auch das Single-Sign-On (SSO) Konzept von *IBM Workplace*, d.h. meldet sich ein Benutzer am *Workplace Server* oder am *Workplace Client* an, so wird er via SSO auch direkt an der Awareness-Komponente angemeldet und erscheint damit im Status „Online“, ohne erneut sein Login und Passwort eingeben zu müssen.

Die Awareness-Funktionalität wird von den verschiedenen *Workplace* Applikationen und Komponenten konsequent eingesetzt. Jeglicher Name von Personen erscheint als Live Name – sowohl in den Komponenten der *Workplace Collaborative Services* Applikationen wie Messaging, Kalender, Teamspaces, Dokumentenbibliothek, Diskussionsforum etc., als auch in den Applikationen im *Workplace Managed Client* inklusive dem Activity-Explorer. Die Verwendung der Live-Names erstreckt sich sogar auf die meisten anderen Produkte der *Workplace* Familie, wie z.B. *Workplace for Business Controls and Reporting*. Allerdings werden Live-Names nur dort verwendet wo Namen als Teile der strukturierten Daten bzw. als Metadaten auftauchen (d.h. also z.B. in der Spalte der Absender in der Mail Inbox, oder als Manager eines Teamspaces im Membership Portlet.) *IBM Workplace* unterstützt allerdings nicht das automatische Erkennen

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

von Personennamen in einem unstrukturiertem Richtext wie z.B. im Body einer E-Mail oder innerhalb eines Textdokumentes in einer Dokumentenbibliothek. Diese Funktionalität wird z.B. von Microsofts SmartTag Technologie bereitgestellt, die unter anderem eine den Live-Names ähnliche Funktionalität in beliebigen Texten wie zum Beispiel in den Microsoft Office Produkten bereitstellt.

Die Awareness-Funktion von *IBM Workplace* basiert, ebenso wie die Instant Messaging und Web-Konferenz Funktion, auf dem „Session Initiation Protocol (SIP)“. SIP ist ein standardisiertes Realtime-Protokoll, welches die Basis jeglicher Realtime-Funktionalität in *Workplace* bildet. Dies ermöglicht unter anderem eine einfache Interoperabilität mit anderen SIP-basierten Produkten anderer Hersteller, z.B. den InstantMessaging Clients von Yahoo, AOL und Google.

Da die Online-Awareness Funktionalität durch das Person-Tag bzw. das Live-Name-Control für den *Workplace Client* realisiert ist, welches im *Workplace Software Development Kit (SDK)* zur Verfügung steht, kann diese Technologie sehr einfach in eigene *Workplace* Komponenten integriert werden.

4.2.6.2 Place-based-Awareness

Im Gegensatz zu der breiten Verwendung der Online-Awareness wird die Place-based-Awareness, welche auch von *Workplace* unterstützt wird, bisher nur in wenigen Komponenten genutzt. Derzeit findet sich die Anwendung von Placed-base-Awareness nur in der Web-Konferenz-Komponente. Das „Teilnehmer“-Portlet (engl. „Attending“) zeigt während der Durchführung einer Web-Konferenz, welche Teilnehmer sich aktuell in die Web-Konferenz „eingewählt“ haben und an ihr teilnehmen. Die Liste der aktuellen Teilnehmer ist ebenfalls mit Live-Names ausgestattet, so dass man neben dem Online-Status auch direkt den Zugriff auf die kollaborativen Funktionalitäten aus dem Kontextmenü des Live-Names hat.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

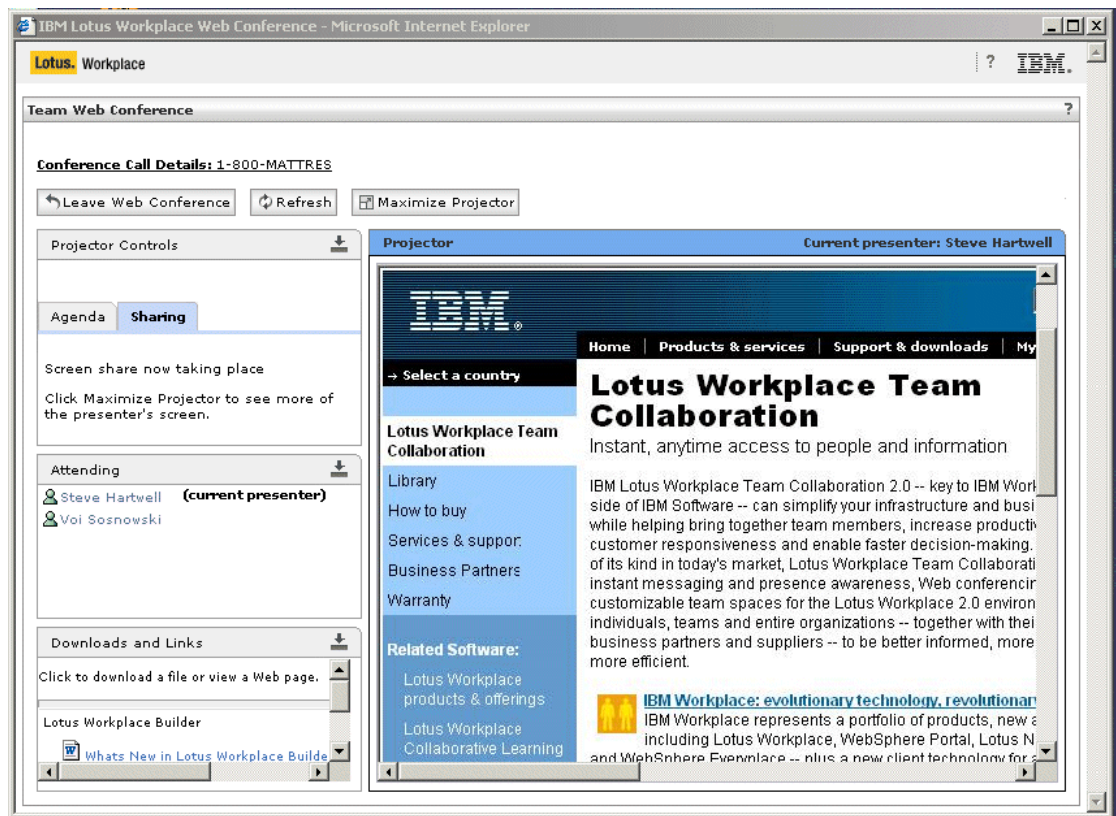


Abb. 28: Place-based-Awareness in einer Web-Konferenz

[IBM c 2006, S.62]

4.2.6.3 Device-based-Awareness

Eine Device-based-Awareness für Personen, wie in Kapitel 3.3.7 beschrieben, ist derzeit nicht in *IBM Workplace* verfügbar.

4.2.6.4 Ressource-Awareness

Das Konzept der Ressource-Awareness findet im Vergleich zur People-Awareness in *IBM Workplace* noch wenig Anwendung. Einzig der Activity-Explorer im *Workplace Managed Client* macht extensiven Gebrauch nicht nur von der People- sondern auch von Ressource-Awareness. Activities enthalten verschiedenste "Shar[ed-Objects", also Dokumente, Chat Transkripts, Dateien, Folder usw., die zur gemeinsamen Nutzung im Team in der Activity abgelegt wurden. „To encourage communication among members, activity explorer goes beyond the use of generalized people awareness by embedding awareness information in every shared object. When any member uses an object, the object is activated. To indicate that an object is active, activity explorer highlights the object in green. Object-level awareness helps to reduce sharing conflicts, and promotes opportunistic collaboration, because a member knows when another member is at work

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

on an object that they have in common, and can coordinate their workflow in real-time.” [IBM b 2006, S.47] Der Activity-Explorer benutzt verschiedene Awareness-Icons um die Ressource-Awareness anzuzeigen. Bearbeitet ein Mitglied der Activity gerade eines der Shared-Objects in der Activity, so wird dessen Objekt Icon grün hinterlegt. So kann jedes andere Mitglied in Echtzeit erkennen, dass ein Objekt gerade von einem anderen Team Mitglied bearbeitet wird. Ist ein Objekt bearbeitet worden, so erscheint das Objekt Icon des bearbeiteten Objektes mit einem kleinen gelben Stern, während die gesamte Activity, in der ein Objekt bearbeitet wurde, mit einem roten Ausrufezeichen markiert wird. So kann man diese Änderung an einem Objekt auch für Activities wahrnehmen die man nicht gerade im Activity-Explorer geöffnet hat, sondern nur in der Activity List sieht.

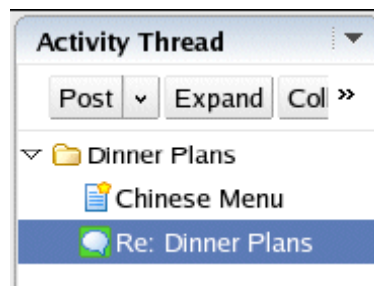


Abb. 29: Ressource-Awareness in einer Activity
[IBM b 2006, S.48]

Bewertung der Kontext-Awareness

In Kapitel 3.3.7 wurde an Kontext-Awareness die Anforderung gestellt, Awareness über Personen als auch über Ressourcen anzuzeigen. Für Personen wurde dabei unterschieden zwischen Online-, Place-based-, und Device-based-Awareness. Darüber hinaus wurde an Kontext-Awareness die Anforderung hergeleitet als funktionale Einheit zur Verfügung zu stehen, um sie in neue Komponenten einbinden zu können.

Die Evaluation von Kontext-Awareness-Funktionalitäten in der *IBM Workplace* Architektur zeigt ein uneinheitliches Bild. Wie beschrieben, steht die Online-Awareness konsequent in allen *Workplace*-Komponenten bereit, sowohl Browser- als auch Desktop-basiert. Diese Komponente wird darüber hinaus vom SDK zur Verfügung gestellt und kann somit in eigene Applikationen integriert werden. Place-based-Awareness wird nur in der Web-Konferenz-Komponente angeboten. Diese Komponente kann derzeit nicht in weitere Komponenten integriert werden da keine API bereit steht. Die Device-based-

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Awareness ist in den derzeitigen *Workplace*-Komponenten nicht implementiert. Die Ressource-Awareness-Funktionalität ist einzig für den Activity-Explorer nutzbar. Sie steht für die Browser-basierte Nutzung nicht bereit. Das SDK stellt darüber hinaus derzeit keine API für die Ressource-Awareness-Funktionalität zur Verfügung.

Abschließend lässt sich sagen, dass die Anforderungen an Kontext-Awareness nur teilweise erfüllt werden. Die *IBM Workplace*-Architektur sieht zwar die Nutzung der unterschiedlichen Awareness-Arten vor, allerdings finden sie bisher noch keine konsequente Nutzung. Als positive Ausnahme ist dabei die Online-Awareness hervor zu heben.

4.2.7 Notification

Wie in Kapitel 3.3.8 Notification beschrieben stellen Notifications eine wichtige Methode dar um den Anwender über wichtige Änderungen in einem Kontext zu informieren, auch wenn er sich gerade außerhalb des entsprechenden Kontextes aufhält. *IBM Workplace* stellt die notwendige Infrastruktur bereit die Notwendig ist um sowohl asynchrone als auch synchrone Notifications versenden zu können. Dies umfasst die *Workplace* Mail Infrastruktur als auch die *Workplace* Instant Messaging-Infrastruktur.

Die bei *IBM Workplace* mitgelieferten Applikationen und Komponenten machen allerdings nur teilweise von der Möglichkeit Gebrauch, Notifications zu versenden. So besteht z.B. die Möglichkeit in den Server basierten *Workplace* Applikationen, Teamspaces und Web-Konferenzen Personen, die neu als Mitglieder zu diesen Spaces hinzugefügt wurden, durch eine E-Mail Notification benachrichtigen zu lassen. Diese Mail-Notifications beinhalten einen Link auf den neu erstellten Space und erlauben es dem Anwender direkt in den neuen Space zu navigieren. Allerdings besteht nicht die Möglichkeit den Anwender über Änderungen an diesen Kontexten zu informieren, z.B. wenn neue Dokumente in dem Kontext erstellt werden oder bestehende Dokumente verändert wurden. Ebenso nutzen die Browser-basierten Komponenten nicht die Möglichkeit Instant Messaging basierte Notifications an den Anwender zu verschicken.

Im Gegensatz dazu macht der *IBM Workplace Client* und insbesondere der Activity-Explorer reichlich Gebrauch von Notifications, insbesondere die Echtzeit-Notifications auf Basis von Instant Messaging.

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur



Abb. 30: Ein Alert an die Mitglieder einer Activity
[IBM b 2006, S.47]

Diese Notifications werden im Activity-Explorer „Alerts“ genannt. Diese Alerts erscheinen als so genannte „Bubbles“, d.h. kleine, nur kurzfristig erscheinende Dialogfenster in der rechten unteren Ecke des Bildschirms. Alerts werden automatisch an alle Mitglieder einer Activity geschickt sobald ein gemeinsam genutztes Objekt in einer Activity von einem anderen Mitglied gerade benutzt wird oder verändert wurde oder wenn neue Mitglieder zu einer Activity oder einem darin enthaltenen Objekt hinzugefügt wurden. Der Anwender hat dabei die Möglichkeit zu konfigurieren, welche Arten von Alerts er erhalten möchte.

Bewertung

IBM Workplace stellt die Infrastruktur bereit, die notwendig ist, um sowohl asynchrone als auch synchrone Notifications versenden zu können. Dies umfasst die *Workplace Mail* Infrastruktur als auch die *Workplace Instant Messaging* Infrastruktur.

Das Versenden von Alerts wird in *IBM Workplace* derzeit nur von Activities genutzt. Es besteht kein generisches Notification-System welches durchgängig z.B. über Änderungen an Dokumenten benachrichtigen könnte, egal in welchem Shared-Space oder in welcher Komponente ein Dokument verändert wird.

Im Rahmen des *Workplace SDK 2.6* wird die *Workplace Mail Messaging API* zur Verfügung gestellt sowie für den *Workplace Managed Client* zusätzlich eine API zum erstellen und versenden von Alerts. So bietet *Workplace* als Plattform alle Voraussetzungen zum Versenden von Mail oder Instant Message basierten Notifications aus selbst entwickelten Komponenten oder Applikationen.

Abschließend lässt sich sagen, dass das Strukturmerkmal „Notification“ für kontextuell-kollaborative Systeme von *IBM Workplace* als Architektur zur Verfügung gestellt wird. Allerdings werden die Möglichkeiten von den mitgelieferten IBM Komponenten, wie dargestellt, noch unzureichend genutzt.

4.2.8 Single-Sign-on

Single-Sign-On ist ein Standarddienst, der von *WebSphere Portal* grundsätzlich realisiert wird. Daher steht dieser Dienst auch für alle *Workplace* Komponenten zur Verfügung. *WebSphere Portal* stellt darüber hinaus für diesen Dienst APIs bereit. Daher kann Single-Sign-On auch für selbstentwickelte *Workplace* Komponenten implementiert werden. Dieses Strukturmerkmal ist somit in der *IBM Workplace* Architektur realisiert.

4.2.9 Kontextnavigation

Das Strukturmerkmal „Kontextnavigation“ wie es in Kapitel 3.3.10 beschrieben wurde, hat die Aufgabe, Kontextbeziehungen, die während der kollaborativen Prozesse entstehen und durch die Strukturmerkmale der dritten Stufe der kontextuellen Kollaboration persistiert werden, sichtbar und navigierbar zu machen. Die Daten der Objekte, die bei der Entstehung und Persistierung von Kontextbeziehungen beteiligt sind, werden als Metadaten bezeichnet. Ein einheitliches Metamodell, welches von allen *Workplace*-Komponenten einheitlich genutzt wird, wie es in Kapitel 3.3.11 beschrieben wird, ist für eine Kontextnavigation als Voraussetzung identifiziert worden.

Die Evaluation hat gezeigt, dass diese beschriebene Art einer Kontextnavigation derzeit nicht implementiert ist. Es lassen sich im UI in keiner der evaluierten *Workplace*-Komponenten Hinweise auf eine Kontextnavigation-Funktion finden. Daher ist das in Kapitel 3.3.10 geforderte Strukturmerkmal „Kontextnavigation“ für ein kontextuell-kollaboratives System derzeit nicht realisiert.

4.2.10 Metamodell

Die *IBM Workplace* Architektur stellt ein grundlegendes Metamodell für die verschiedenen kollaborativen Funktionen und Komponenten bereit, die mit *IBM Workplace* ausgeliefert werden. Dieses Metamodell ist als Teil des *Workplace* SDK dokumentiert und damit auch für andere Entwickler und Dritthersteller nutzbar. Allerdings deckt das

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Workplace Metamodell durch die Basisklasse „WorkplaceObject“ und der verschiedenen daraus abgeleiteten Klassen wie „Document“, „MailMessage“ oder „Discussion-Post“ vor allem die verschiedenen Objekte und Objekttypen des in Kapitel 3.3.11 definierten Metamodells des idealtypischen Frameworks ab.[vgl. IBM e 2006, S.18] Das *IBM Workplace* Metamodell definiert allerdings nicht die für die kontextuelle Kollaboration wichtigen Elemente wie „Referenz“, „Beziehung“ oder „Kontext“. Es bleibt damit Aufgabe der jeweiligen Komponenten diese Mechanismen individuell zu implementieren, was der konsistenten, plattformweiten Nutzung, Persistierung und schließlich auch Navigation von Objektreferenzen und Beziehungen im Kontext der Geschäftsprozesse im Wege steht.

Die im idealtypischen Framework definierten Anforderungen an das Systemmerkmal „Metamodell“ sind damit nur zum Teil erfüllt.

4.3 Zusammenfassende Bewertung

Nachstehend werden die Evaluationsergebnisse, die in Kapitel 4.2 beschrieben und bewertet wurden, noch einmal tabellarisch zusammengefasst. Dabei wird jeweils unterschieden, ob ein Merkmal von den *Workplace Client*-basierten bzw. den Browser-basierten Applikationen genutzt wird, und ob es im Rahmen der *Workplace* Architektur von Entwicklern und Drittherstellern zur Entwicklung eigener *Workplace Client* bzw. Browser-basierten Komponenten zur Verfügung steht.

Tabelle 3: Zusammenfassende Bewertung der IBM Workplace Architektur

Strukturmerkmale	Desktop-basiert realisiert (WMC)	Stehen Entwicklern für zur Erweiterung zur Verfügung	Browser-basiert realisiert	Stehen Entwicklern für zur Erweiterung zur Verfügung
Stufe 1: Starten kollaborativer Funktionen aus dem Arbeitskontext.				
Komponentisierung	Realisiert durch Plugin-Modell	Komponentenmodell lässt Erweiterungen zu	Realisiert durch Portlets	Komponentenmodell lässt Erweiterungen zu
Komponentenintegration				
API-basiert	realisiert	-	realisiert	
Komponenten--basiert		-	realisiert	
Objekt-basiert	Realisiert für Personen und weitere Objekttypen	Für Personen	Realisiert für Personen	Für Personen
Single-Sign-On	realisiert	Ja	realisiert	ja
Stufe 2: Einbinden des Arbeitskontextes in die Kollaboration				
Interkomponentenkommunikation	Realisiert, allerdings keine Definition vom Anwender möglich	möglich auf API-Ebene als auch durch Eclipse Extension-Points	teilweise realisiert, wird von <i>Workplace</i> -Portlets kaum genutzt	Realisiert auf API-Ebene und auf Komponenten ebene (Property-Broker)

4. Evaluation: Umsetzung der Konzepte von contextual collaborative frameworks am Beispiel der IBM Workplace-Architektur

Stufe 3: Persistierung der Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen				
Kontext-Sharing				
Ad-hoc	Realisiert durch den Activity-Explorer	Nicht erweiterbar	Nicht realisiert	-
Semi-formale	Realisiert durch Teamspaces		Realisiert durch Teamspaces	Erweiterbar als Composite-Application
Formale, workflow-getriebene	Nicht realisiert	-	Nicht realisiert	-
Mitgliedschaft	realisiert	Ja	realisiert	ja
Kontext-Awareness				
Online-Awareness	realisiert	Ja	realisiert	ja
Place-based	Teilweise realisiert (Web-Conferencing)	Nein		nein
Device-based	nein	-	nein	-
Ressource	Teilweise realisiert (im Activity-Explorer)	Nein	nein	-
Notification	teilweise	Ja	teilweise	ja
Metadatenmodell		Ja		ja
Stufe 4: Navigation durch das multidimensionale Geflecht von Kontextbeziehungen				
Kontextnavigation	Nicht realisiert	-	Nicht realisiert	-

5. Ausblick

Im Rahmen der vorliegenden Diplomarbeit werden viele Aspekte des Kollaborationskonzeptes „kontextuelle Kollaboration“ beleuchtet.

Das Prinzip der Anpassung an den derzeitigen Arbeitskontext kann auch für andere Bereiche als die Kollaboration betrachtet werden. So ist es vorstellbar, dass ein System Kontextinformationen über das Arbeitsverhalten von Personen sammeln und auswerten kann. Z.B. könnte es erkennen dass eine Person beim ersten „Betreten“ des Systems bei Arbeitsbeginn die Inbox öffnet und den Teamkalender abgleicht und überprüft. Dieses Verhalten zu erkennen, es auf die Person kontextualisieren zu können und dementsprechend zu reagieren, kann für einzelne Anwender, aber auch für Teams eine Verbesserung der Arbeitsunterstützung bedeuten. Diese beschriebenen Systeme fallen in das Forschungsgebiet von „Contextual Computing“ oder auch „ubiquitous Computing“ [vgl. Prekop/Burnett 2003, S.1-3]

Ein weiterer in diesem Zusammenhang zu betrachtender Aspekt sind die Kosten für Unternehmen bei der Umsetzung und Durchführung von kontextueller Kollaboration: „And there are questions of cost. Pilot programs and trials do not accurately reflect the final total cost of ownership for real-time collaboration, and until companies can predict the impact implementations will have their budgets, moving ahead will be difficult.“ [Perey 2003, S.3] Eine wichtige Untersuchung im Rahmen dieses Themenkomplexes ist es, eine gezielte Effizienzanalyse durchzuführen, zwecks einer besseren Einschätzung der tatsächlichen Kosten/Nutzenverhältnisse der kontextuellen Kollaboration.

Im Rahmen dieser Arbeit wurde kontextuelle Kollaboration vor allem unter dem Aspekt der innerbetrieblichen Kollaboration und Kommunikation betrachtet. In wieweit sich kontextuelle Kollaboration in vollem Umfang auf interorganisationelle Kollaboration ausweiten lässt, bleibt zu untersuchen. Besonders die Aspekte von Sicherheit der eigenen Unternehmensressourcen und Trust-Mechanismen sind dabei zu bedenken.

Es wird im Rahmen dieser Arbeit die besondere Bedeutung von Kontextbeziehungen und dem daraus entstehenden multidimensionalen Beziehungsgeflecht diskutiert. Die Kontextnavigation wird als eine direkte Anwendung für die kontextuelle Kollaboration identifiziert. Um die Kontextnavigation in vollem Umfang implementieren zu können, bedarf es der Definition eines einheitlich verwendbaren komponentenübergreifenden

Metamodells. Die Erstellung eines solchen Modells ist somit ein weiter reichendes Forschungsfeld.

Es gibt in diesem Zusammenhang weitere, im Rahmen der Arbeit nicht betrachtete Anwendungen. Die Analysemöglichkeiten eines solchen persistierten Beziehungsnetzes können für Unternehmen einen Informationsvorsprung bedeuten. Zu wissen welche Personen im Team auf welche Weise kommunizieren und kollaborieren, welche Teamzusammensetzungen besonders effizient arbeiten oder an welcher Stelle in dem Netzwerk Engpässe im Sinne von Entscheidungsrückstaus sind, sind dabei nur wenige Beispiele der Informationen, die solche Analysen liefern können. Diese Analysen fallen unter das Forschungsgebiet der „Social Network Analysis“.

Die in dieser Arbeit als Konzept beschriebenen und auch bei IBM Workplace eingesetzten „Live Names“ bieten eine sehr gut integrierte Form der kontextuellen Kollaborationsfunktion. Dieses Konzept kann und sollte auch auf andere Objekttypen als nur Personen erweitert werden, z.B. auf Dokumente. Die auf ihnen ausführbaren Funktionen werden entsprechend ihrem Benutzungskontext angezeigt und ausgeführt. Ein Anwendungsbeispiel ist die Funktion „Versende als Email“ direkt aus dem Kontext des Dokumentes

6. Zusammenfassung

„Kontextuelle Kollaboration“ ist ein besonderes Kollaborationskonzept, das die Kollaboration im Kontext von Geschäftsprozessen in den Mittelpunkt stellt um so das Prozessergebnis zu verbessern. Das Konzept wurde zunächst begrifflich erfasst und definiert. Im Rahmen dieser Arbeit wurden die Anforderungen an kontextuell-kollaborative Systeme spezifiziert und analysiert. Dazu wurden zunächst die Konzepte der kontextuellen Kollaboration untersucht und systematisiert. Als Ergebnis dieser Konzeptanalyse konnte ein vierstufiges Modell für die kontextuelle Kollaboration entwickelt werden. Es wird im Rahmen dieser Arbeit als „**Vier-Stufenmodell der kontextuellen Kollaboration**“ bezeichnet. Dieses Modell beschreibt vier aufeinander aufbauende Integrationsstufen kollaborativer Funktionalität mit jeweils wachsender Mächtigkeit der Kontextualisierung. Diese Stufen sind:

1. Starten kollaborativer Funktionen aus dem Arbeitskontext
2. Einbinden des Arbeitskontextes in die Kollaboration
3. Persistierung von Beziehungen zwischen Kollaborationsergebnissen und Geschäftsprozessen
4. Navigation durch das multidimensionale Geflecht von Kontextbeziehungen

In besonderer Weise werden dabei die bei der Kollaboration entstehenden Kontextbeziehungen betrachtet. Es wurde gezeigt, in welcher Form sie ein multidimensionales Beziehungsgeflecht aufspannen und dass sie durch ein einheitliches Metadatenmodell repräsentiert werden können.

Ein praxisnahes Use-Case-Szenario zeigte die innovative und flexible Arbeitsweise mit einem kontextuell-kollaborativen System und verdeutlicht insbesondere, welche Vorteile sich für Anwender aus dieser Form der Kollaboration ergeben. Das Use-Case-Szenario zusammen mit dem „Vier-Stufenmodell der kontextuellen Kollaboration“ diente als Grundlage für die Spezifikation von Anforderungen, die an ein solches System gestellt werden müssen. Aus diesen identifizierten Anforderungen ließ sich ein Framework erstellen, unter dem besondern Aspekt der Funktionen, die ein kontextuell-kollaboratives System in besonderer Weise unterstützen. Diese Funktionen und Konzepte werden im Rahmen dieser Arbeit „Strukturmerkmale“ genannt. Die Strukturmerkmale konnten den vier Stufen des „Vier-Stufenmodells“ zugeordnet werden. Die Bedeutung, Wirkungsweise und Eigenschaften der Strukturmerkmale in kontextuell-kollaborativen Systemen wurden intensiv diskutiert und erläutert. Aus der Definition

dieser Merkmale wurde ein idealtypisches Framework für kontextuell-kollaborative Systeme entwickelt, welches zusammen mit der Systematik des Vier-Stufenmodells grundsätzlich geeignet ist, kollaborative Systeme auf den Grad ihrer kontextuellen Kollaboration zu evaluieren und zu klassifizieren.

Die im Rahmen dieser Arbeit aufgestellten und als Strukturmerkmale beschriebenen Kriterien für ein kontextuell-kollaboratives Framework sind dann zur Grundlage genommen worden, um die *IBM Workplace* Architektur zu evaluieren. Die Bewertung der Umsetzung der einzelnen Strukturmerkmale ist in den Abschnitten der einzelnen Evaluationsschritte erläutert.

Die *IBM Workplace* Architektur zeigt sich grundsätzlich als geeignete Architektur, um kontextuell-kollaborative Systeme zu realisieren. Viele der Strukturmerkmale sind vollständig umgesetzt, wie z.B. „Komponentisierung“ und die unterschiedlichen Formen der „Komponentenintegration“. Auch bei den teilweise realisierten Strukturmerkmalen zeigt sich, dass sie grundsätzlich in der Architektur vorgesehen sind, aber noch nicht konsequent von *IBM Workplace* selber genutzt werden bzw. noch nicht für eigenentwickelte Komponenten zur Verfügung stehen, Beispiele hierfür sind die Ressource-Awareness oder die objektbasierte Integration kollaborativer Funktionen für Dokumente und andere Objekttypen. Es hat sich außerdem gezeigt, dass einige der Funktionalitäten nicht konsequent auf beiden Client-Plattformen, dem *Workplace Managed Client* sowie dem Browser-basierten Portal-Client realisiert sind, wie z.B. der Activity-Explorer.

Die Autorin sieht aufgrund der dargestellten Ergebnisse der Evaluation die *IBM Workplace* Architektur in weiten Teilen als geeignet, um ein kontextuell-kollaboratives System in einem Unternehmen zu realisieren. Als Portal-basierte Integrationsplattform scheint sie in besonderer Weise die Kernkonzepte der kontextuellen Kollaboration umsetzen zu können. „Portal Frameworks remain the best method for unifying delivery of collaboration ‘in context’ with other content, applications, and workflow needs (e.g. role, rules). This movement towards contextual collaboration (defined by META Group in 1999) continues to be adopted by major technology vendors (e.g. IBM, Microsoft) and Global 2000 clients to improve productivity, reduce coordination costs, and better connect people to peers and teams.” [Gotta, 2004, S.3] *IBM Workplace* zeigt sich als eine innovative Integrationsplattform. Durch ihr besonderes Komponentenmodell, das sowohl *Eclipse Rich Client*-basierte Plugin-Komponenten unterstützt als auch Portlets, wird sie die Plattform vor allem für Erweiterungen interessant und gut einsetzbar. Durch

die möglichen Erweiterungen können so kontextuell-kollaborative Systeme auf Basis von IBM Workplace entwickelt werden. Die im Rahmen dieser Arbeit entwickelten Modelle und die herausgearbeiteten Strukturmerkmale dienen dafür als Leitfaden.

Anhang A

Kurzbeschreibung einiger IBM Workplace Produkte:

IBM Workplace Software Development Kit (SDK) dient Entwicklern dazu die *Workplace* Plattform zu nutzen und zu erweitern, z.B. durch neue Portlets oder Business-Komponenten. Das SDK liefert eine Reihe von public APIs und SPIs. Sie ergänzen die public APIs des zugrunde liegenden *WebSphere Applications Servers* und des *WebSphere Portal Servers*. SDK umfasst die folgenden APIs, SPIs und JSP tags: Collaborative Application Component Interfaces, Component and Application Infrastructure Services APIs, Mail Messaging SPI for *IBM Workplace Collaborative Services*, Instant Messaging SPI for *IBM Workplace Server Products*, JSP tags for *IBM Workplace Server Products*

[[http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcs26api/\\$File/wcsapi26_ug.pdf](http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcs26api/$File/wcsapi26_ug.pdf)]

IBM Workplace for Business Strategy Execution ist eine Komponente, die Führungskräfte in der Ausführung der unternehmensspezifischen Business-Strategien unterstützt. Sie ist allerdings keine Software, die Business-Strategien entwirft. [<http://publib.boulder.ibm.com/infocenter/wbsehelp/v1r0/index.jsp>]

IBM Workplace Forms ist eine weitere Komponente der Produktfamilie, die Funktionalitäten rund um die Erstellung und Verwaltung von XML Form basierten Prozessen und für die Entwicklung von Standard basierten Forms bereitstellt. [McCarrick, D.: *Introducing IBM Workplace Forms*, 2005 IBM Developerworks <http://www-128.ibm.com/developerworks/workplace/library/workplace-forms/>]

IBM Workplace Services Express ist eine schlankere Version der *IBM Workplace Collaboration Services*. Mit der Zusammenstellung der Funktionen zu diesem Paket adressiert IBM vor allem den Markt der kleineren und mittelgroßen Unternehmen (SMB-Markt). Die Kollaborationsfunktionalitäten sind auf diese Zielgruppe abgestimmt. [<http://publib.boulder.ibm.com/infocenter/wseic/v2r5/index.jsp>]

IBM Workplace Dashboard bietet wiederverwendbare Komponenten zur Erstellung Anzeige und Auswertung so genannter Dashboards. Die Dashboards sind Komponenten zur Überwachung und Steuerung unternehmensrelevanter Kennzahlen, welche dynamisch erzeugt werden, wie z.B. Verkaufszahlen oder der Lagerbestand. [ftp://ftp.software.ibm.com/software/lotus/lotusweb/workplace/dashboards/Dashboard_Framework_data_sheet.pdf]

Bei dem **IBM Workplace Designer** handelt es sich um ein Werkzeug zur Entwicklung von *Workplace* Komponenten und deren Veränderung und Anpassung. Diese entwickelten Komponenten lassen sich den *Workplace* Applikationen hinzufügen und mit einem Web Browser nutzen. (“IBM® Workplace™ Designer 2.6 is a development tool for creating components that can be added to IBM Workplace applications and used with a Web browser. Workplace Designer provides a Tool for creating the components, and a Runtime that allows the components to be deployed as applications on the server.” [[http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcs26/\\$File/r_inst_designer.pdf](http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcs26/$File/r_inst_designer.pdf)]).

Literaturverzeichnis

[Advanced Reality 2002]

Contextual Collaboration: The key to boosting processes Advanced Reality inc. 2002
http://www.advancedreality.com/ar/ressources/Items/contextual_collaboration.pdf
letzter Zugriff 12.2.2006

[Apelt 2003]

Apelt, D.: Single Sign-On: An Overview To Integration, Whitepaper; 2003; Protocom
Development Systems 2003
[http://www.protocom.com/html/whitepapers/tech_sso_integration.html]

[Bargas 2005]

Bargas, D.: IBM Workplace Client Technology architecture Februar 2005 IBM
developerworks 2005
<http://www-128.ibm.com/developerworks/lotus/library/wct-architecture/> letzter Zugriff
11.04.2006

[Boyd 2003]

Boyd, S.: Collaboration in Context 2003
[<http://www.destinationkm.com/articles/default.asp?ArticleID=1103>] letzter Zugriff
10.4.2006

[Bruse 2005]

Bruse, T.: Groupware-basiertes Workplace-Portal G8: Konzeption und prototypische
Implementierung eines Workplace-Portals zur Unterstützung des organisatorischen
Wissensmanagements, Paderborn, Univ., Diss., 2005

[Cain 2003]

Cain, M.: Mapping Collaboration Maturity , November 21 , 2003
[http://techupdate.zdnet.com/techupdate/stories/main/Mapping_Collaboration_Maturity.html]

[Chen et.al. 2004]

Chen, T.; Kekre, P.; Lan, T.; Lee, J.; Lin, A.; Soni, B.: Contextual Collaboration, a
study in virtual team communication with relation to global corporate development;
Carnegie Mellon MISN; information systems project Dezember 2004
<http://www.mism.cmu.edu/futureStudents/studentProjects/Contextual%20Collaboration%20White%20Paper.pdf> letzter Zugriff 12.4.2006

[Coleman, D. 2002]

Coleman, D. 2002 Levels of Collaboration, Collaborative Strategies 2002

http://www.collaborate.com/publication/newsletter/publications_newsletter_march02.html

[Cross et al. 2001]

Cross, R., Parker, A., Prusak, L., & Borgatti, S.P: Knowing What We Know: Supporting Knowledge Creation and Sharing in Social Networks. *Organizational Dynamics* 30 (2): 100 – 120; 2001]

[Dey/Abowed 1999]

Dey, A.K.; Abowed, G.D.: Towards a Better Understanding of Context and Context-Awareness; In the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands, April 3, 2000

Also GVU Technical Report GIT-GVU-99-22. Submitted to the 1st International Symposium on Handheld and Ubiquitous Computing (HUC '99), June 1999

[Dierker/Sander 1998]

Dierker, M.; Sander, M.: Lotus Notes 4.6 und Domino: Integration von Groupware und Internet; Addison-Wesley, Bonn et al. 1998

[Dourish/Bellotti 1992]

Dourish, P.; Bellotti, V.: Awareness and Coordination in Shared Workspaces; Computer Supported Cooperative Work archiveProceedings of the 1992 ACM conference on Computer-supported cooperative work; Toronto, Ontario, Canada ;Pages: 107 - 114 ;Year of Publication: 1992

[Duden 2004]

Duden: Wörterbuch der deutschen Sprache in 10 Bänden. Aktualisierte Online-Ausgabe. Mannheim, Leipzig, Wien, Zürich: Dudenverlag 1999-2004.

[Ehrlich/Carboni 2005]

Ehrlich, K.; Carboni, I.: Collaboration, Social networks, SNA, knowledge management, Copyright 2005, IBM. Watson research Center Technical Report 05-10, UCINet, 2005 <http://domino.watson.ibm.com/cambridge/research.nsf/2b4f81291401771785256976004a8d13/3f23b2d424be0da6852570a500709975?OpenDocument> letzter Zugriff 10.4.2006

[Fischer et al. 2003]

Fischer, P. Hesmer, S. Schaeck, T.: Understanding the Portlet Component Model in IBM WebSphere Portal IBM Software Group July 2003

<ftp://ftp.software.ibm.com/software/dw/wes/pdf/WebSpherePortalandPortlets.pdf>

[Gamma et al.1995]

Gamma, E.; Helm,R.; Johnson, R.; Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software, 1995, Addison Wesley, Reading, MA]

[Gamma/Beck 2004]

Gamma, E.; Beck, K. : Eclipse erweitern – Prinzipien, Patterns und Plugins, Addison-Wesley 2004, München

[Gesellschaft f. Informatik 2006]

Gesellschaft für Informatik Fachgruppe CSCW <http://wwwfgcsw.in.tum.de/csw-collaboration.html> letzter Zugriff 10.4.2006

[Gotta 2004]

Gotta, M.: Collaboration: The Future Is Contextual, Process-Centric and Community-Driven; META-Group; May 2004 2004

http://techupdate.zdnet.com/techupdate/stories/main/Collaboration_Future_Is_Contextual.html letzter Zugriff 10.4.2006

[Groeger/Gorman 2004 a]

Groeger, H. Gorman, M.: IBM Workplace application development: IBM Workplace as a collaborative Application framework, 13.Dec.2004

<http://www-128.ibm.com/developerworks/lotus/library/lwp-appdev1/> letzter Zugriff: 30.3.2006

[Groeger/Gorman 2004 b]

Groeger, H. Gorman, M.: IBM Workplace application development: IBM Woerklpace programming model 20.Dec. 2004

<http://www-128.ibm.com/developerworks/lotus/library/lwp-appdev2/> letzter Zugriff: 10.4.2006

[Gruen 2000]

Gruen, D.: Storyboarding for Design: An Overview of the Process; Technical Report 2000.03; IBM Research

<http://domino.research.ibm.com/cambridge/research.nsf/2b4f81291401771785256976004a8d13/ebcd159a81a43e36852569200067d59e?OpenDocument> letzter Zugriff 11.4.2006

[Hahl 2004]

Hahl, O.: Föderierte Portale zur Überwindung inner- und zwischenbetrieblicher Portalproliferation; 2004 K-Pool GCC

[Hall 1999]

Hall, T.: Intelligence Community Collaboration, Baseline Study Final Report, 1999

http://collaboration.mitre.org/prail/IC_Collaboration_Baseline_Study_Final_Report/toc.htm letzter Zugriff 10.4.2006

[Herzum/Sims 2000]

Herzum,P.; Sims, O.: Business Component Factory; John Wiley&Sons, Inc., NewYork et al. 2000

[Huang, G. 2004]

Huang, G.: Wissensmanagement: Wissensversorgung und kontextuelle Kollaboration bei Workflow-basierter Geschäftsprozessabwicklung; Dissertation Shaker Verlag: Aachen, 2004

[Hupfer et al. 2004]

Hupfer, S.; Cheng, L.T.; Ross, S.; Patterson, J.: Introducing Collaboration into Application Development Environment; CSCW 2004 - ACM 2004 Conference on Computer Supported Cooperative Work November 6-10, 2004 Chicago, Illinois, USA; IBM Research Collaborative User Experience Group Copyright 2004, IBM

[Hupfer et.al. 2005]

Hupfer, S.; Cheng, L.T.; Ross, S.; Patterson, J.: Reinventing Team Spaces for a Collaborative Development Environment; submitted to: "beyond Threaded Conversation" Workshop CHI 2005 April 2005 Copyright IBM 2005

[IBM InfoCenter a 2006]

IBM InformationCenter :Overview of cooperative Portlets

<http://publib.boulder.ibm.com/infocenter/wpdoc/v510/topic/com.ibm.wp.ent.doc/wps/wpsc2acnc.html> letzter Zugriff: 9.April 2006

[IBM a 2006]

IBM Workplace Collaboration Services, Version 2.6; Information Center 2006

<http://publib.boulder.ibm.com/infocenter/jwphelp/v2r6m0/index.jsp?toc=/com.ibm.wcs26.ic.doc/toc.xml> letzter Zugriff 31.03.2006

[IBM b 2006]

IBM Workplace Managed Client on Linux 2.6

<http://www.redbooks.ibm.com/redbooks.nsf/65f0d9cea6e0ab57852569e0007452bb/f5abf55e5437be3d852570c700143b50?OpenDocument> letzter Zugriff: 19.April 2006

[IBM c 2006]

IBM Lotus Workplace Team Collaboration 2.0.1

<http://www.redbooks.ibm.com/abstracts/redp3929.html?Open> letzter Zugriff: 19.April 2006

[IBM d 2006]

IBM Workplace Services Express

<http://www.redbooks.ibm.com/abstracts/sg246758.html?Open> letzter Zugriff: 19.April 2006

[IBM e 2006]

IBM Workplace Software Development Kit Version 2.6 User Guide

<http://www-142.ibm.com/software/workplace/products/product5.nsf/wdocs/apitoolkit>
letzter Zugriff: 19.April 2006

[ITWissen]

http://www.itwissen.info/definition/lexikon//e-mail__electronic%20mail_elektronische%20post.html

[Iyengar et.al. 2005]

Iyengar, A.K.; Gadepalli, V.V.; Olson, B.R.: IBM Websphere Portal Primer; IBM Press Lewisville, Texas 2005

[Leitner/Tuppinger 2004]

Leitner, W.; Tuppinger, J. 2004: "Wissenstransfer bei der virtuellen Teamarbeit" - Buchbeitrag in: Engelhardt, C.; Hall, K.; Ortner, J. (Hrsg.): Prozesswissen als Erfolgsfaktor - Effiziente Kombination von Prozessmanagement und Wissensmanagement, Wiesbaden 2004

[IT-Fachbegriffe]

Lexikon IT-Fachbegriffe <http://www.interest.de/cgi-bin/lexika/VoIP.html?pos=T15791628&ID=857454019843> Stichwort „VoIP“ letzter Zugriff 14.4.2006

[Lotosphere 2006]

Lotosphere Recap Praesentation, Lotusphere 2006,

[http://gcc.upb.de/www/WI/WI2/wi2_lit.nsf/1e0465add2cf1292c1256f3c003fe9fe/e7ae9e43288adbca12571170043ed1a/\\$FILE/Lotosphere%20Recap%202006.pdf](http://gcc.upb.de/www/WI/WI2/wi2_lit.nsf/1e0465add2cf1292c1256f3c003fe9fe/e7ae9e43288adbca12571170043ed1a/$FILE/Lotosphere%20Recap%202006.pdf)

[Mahowald 2001]

Mahowald, R.: From ICE Age To Contextual Collaboration; IDC Report, CXO Media Inc., Juni 29, 2001

[http://www.cio.com/analyst/062901_idc.html] letzter Zugriff 10.4.2006

[Millen et al. 2005]

Millen, D.R.; Muller, M.J.; Ehrlich, K.: Understanding Patterns of Media Use in an Activity-Centric Collaborative Environment; Proceedings of the SIGCHI conference on Human factors in computing systems Portland, Oregon, USA SESSION: Understanding users and usage patterns Pages: 879 - 888 Year of Publication: 2005

[Mitchell 1998]

Mitchell, D.: A Component Approach to Embedding Awareness and Conversation; Presented at WETICE '98; the Seventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, June 1998

[Monson et.al. 2005]

Monson, P.; Clark, J.; Mikkolainen, K.; Shalabi, S.: Building a Component for IBM Workplace IBM Redpaper; Copyright IBM Corp. 2005

<http://www.redbooks.ibm.com/redpapers/pdfs/redp3952.pdf> letzter Zugriff 30.03.2006

[Moran 2005]

Moran, T.P.; Cozzi, A.; Farrell, S.P.: Unified Activity Managemet: Supporting People in eBusiness; Communications of the ACM, Special Section on Semantic eBusiness Vision; December 2005

[Muller et.al. 2004]

Muller, M. J.; Geyer, W.; Brownholtz, B.; Wilcox, E.; Millen, D.R.: One-Hundred Days in an Activity-Centric Collaboration Environment based on Shared Objects; CHI 2004, April 24-29, 2004, Vienna, Austria Copyright ACM 2004 IBM Research

[Nardi 1996]

Nardi, B.A.: Studying Context: A Comparison of Activity Theory, Situated Action Models, and Distributed Cognition; In: Context and Consciousness: activity theory and human-computer interaction; Nardi, B.A. 1996 The MIT Press, USA 1996

[Nastansky et. al. 2000]

Nastansky, L. et.al.: Büroinformations- und Kommunikationssysteme: Groupware, Workflow-Management, Organisationsmodellierung und Messaging-Systeme; in: Fischer, J. et. al. Bausteine der Wirtschaftsinformatik; Erich Schmidt Verlag 2000

[Oesterle et al. 2002]

Österle, H.; Fleisch, E.; Alt, R.: Business Networking in der Praxis: Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten; Springer, Berlin 2002

[Perey 2003]

Perey, Ch.: Real-time collaboration – for real! Network World 09/29/03; 2003 <http://www.networkworld.com/buzz/2003/0929collab.html> letzter Zugriff 11.4.2006

[Schiffrin 1994]

Schiffrin, D.: Approaches to Discourse. Oxford: Blackwell 1994

[Schmitz 1991]

Schmitz, G.: Methode und Techniken der Organisation, Schriftreihe „Der Organisator“ Band 1, Verlag Dr. Götz Schmitz, Gießen 1991]

[Shalabi 2006]

Shalabi, S.: IBM Workplace programming model overview 04.Jan. 2006

<http://www-128.ibm.com/developerworks/workplace/library/workplace-programming-model/>

[Silva Filho et.al. 2005]

Silva Filho, R.S.; Geyer, W.; Brownholtz, B.; Guy, I.; Redmiles, D.F.; Millen; D.R.: Architectural Trade-Offs for Collaboration Services Supporting Contextual Collaboration; 2005, IBM T.J. Watson Research Center 2005

[Sparling 2000]

Sparling, M.: Lessons learned Through six years of component-based development in: communications of the ACM Vol.43, No.10 Oktober/2000]

[ter Hofte et al. 2002]

ter Hofte, H.G.; Mulder, I. Grootvelt, M.; Slagter, Robert: Beyond being online: Exploring place-based presence; presented at the CSCW Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, held in conjunction with CSCW 2002, the ACM 2002 Conference on Computer Supported Cooperative Work, New Orleans, LA, USA, November 16-20, 2002.

[Wegener 2002]

Wegener, H.: Analyse und objektorientierter Entwurf eines integrierten Portalsystems für das Wissensmanagement Copyright dissertation.de – Verlag im Internet GmbH Berlin <http://www.dissertation.de> 2002

[WfMC 1999]

Workflow Management Coalition (ed.): Terminology and Glossary Document Number WFMC-TC-1011 1999

http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf letzter Zugriff 10.4.2006