



UNIVERSITÄT
KOBLENZ · LANDAU

Institut für Wirtschafts-
und Verwaltungsinformatik



FB 4

Informatik

Integration von CRM-Systemen mit Kollaborations-Systemen am Beispiel von DocHouse und Lotus Quickr

Martin Surrey et al.

Nr. 3/2012

**Arbeitsberichte aus dem
Fachbereich Informatik**

Die Arbeitsberichte aus dem Fachbereich Informatik dienen der Darstellung vorläufiger Ergebnisse, die in der Regel noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar. Alle Rechte vorbehalten, insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

The “Arbeitsberichte aus dem Fachbereich Informatik“ comprise preliminary results which will usually be revised for subsequent publication. Critical comments are appreciated by the authors. All rights reserved. No part of this report may be reproduced by any means or translated.

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346

ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:
Prof. Dr. Grimm

Die Professoren des Fachbereichs:

Prof. Dr. Bátori, Prof. Dr. Burkhardt, Prof. Dr. Diller, Prof. Dr. Ebert,
Prof. Dr. Furbach, Prof. Dr. Grimm, Prof. Dr. Hampe, Prof. Dr. Harbusch,
jProf. Dr. Kilian, Prof. Dr. von Korflesch, Prof. Dr. Lämmel, Prof. Dr. Lautenbach,
Prof. Dr. Müller, Prof. Dr. Oppermann, Prof. Dr. Paulus, Prof. Dr. Priese,
Prof. Dr. Rosendahl, jProf. Dr. Scherp, Prof. Dr. Schubert, Prof. Dr. Sofronie-
Stokkermans, Prof. Dr. Staab, Prof. Dr. Steigner, Prof. Dr. Sure, Prof. Dr. Troitzsch,
Prof. Dr. Wimmer, Prof. Dr. Zöbel

Kontakt Daten der Verfasser

Björn Lilge, Ludwig Paulsen, Marco Wolf, Markus Aldenhövel, Martin Surrey, Mike Reuther, Roland Diehl
Institut für Wirtschafts- und Verwaltungsinformatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
E-Mail: blilge@uni-koblenz.de, lpaulsen@uni-koblenz.de, marcowolf@uni-koblenz.de, maldenhoevel@uni-koblenz.de, msurrey@uni-koblenz.de, mreuther@uni-koblenz.de, rdiehl@uni-koblenz.de

Vorwort und Danksagung

CRM-Systeme als Teilmenge von betrieblichen Anwendungssystemen bilden einen wichtigen Kompetenzbereich der Forschungsgruppe Betriebliche Anwendungssysteme (FG BAS) am Institut für Wirtschafts- und Verwaltungsinformatik (IWVI), der Universität Koblenz-Landau. Die Forschungsgruppe vermittelt in diesem Themengebiet sowohl Ergebnisse aus ihrer Grundlagenforschung als auch praxisbezogene Inhalte an Studierende und Unternehmen. Die enge Zusammenarbeit mit Partnern aus Wirtschaft und Wissenschaft gewährleistet hier den Praxisbezug der Forschungsergebnisse.

Der vorliegende Arbeitsbericht „Integration von CRM-Systemen mit Kollaborations-Systemen am Beispiel von DocHouse/ BRM und IBM Lotus Quickr“ ist Ergebnis einer studentischen Projektarbeit. Ziel des Projekts war es Integrationsszenarien zwischen einem CRM-System und einem Kollaborationssystem zu erarbeiten und eine prototypische Schnittstelle mit entsprechender Funktion zwischen den Systemen DocHouse/ BRM und IBM Lotus Quickr zu implementieren. Ein besonderer Dank geht in diesem Zusammenhang an Herr Wolfgang Brugger (Geschäftsführer der DocHouse GmbH), der die Idee einer solchen Entwicklung hatte und die FG BAS mit deren Durchführung betraute. Die Erstellung des Konzepts und des Prototyps wurde vom Winter 2010 bis Sommer 2011 von den Studenten Björn Lilge, Ludwig Paulsen, Marco Wolf, Markus Aldenhövel, Martin Surrey und Mike Reuthers im Rahmen ihres Projektpraktikums durchgeführt. Das Projektteam wurde bei der Konzeption und Implementierung inhaltlich und organisatorisch von Dipl.-Wirt.-Inform. Roland Diehl betreut.

Die Forschungsgruppe von Frau Prof. Dr. Schubert ist auf die Durchführung solcher Projekte spezialisiert. Jährlich werden Projektpraktika für Studenten angeboten. Oftmals werden Prototypen als Ergebnis dieser Projektpraktika erstellt. Die Wirtschaftspartner profitieren bei diesen Projekten von der sachorientierten und unabhängigen Sicht der FG BAS, vom Know-How und den Kapazitäten bei der Durchführung der Projekte, bei der Systematisierung und Aufbereitung von Inhalten und bei der Vermittlung von Wissen über firmenunabhängige Kanäle.

An dieser Stelle geht ein Dank an die Studenten des Projektteams, die so freundlich waren, die Ergebnisse ihres Projektpraktikums in einer umfassenden Form als Arbeitsbericht des Fachbereichs Wirtschafts- und Verwaltungsinformatik zu veröffentlichen. Dank gilt auch den wissenschaftlichen Mitarbeitern, die für die Bereitstellung der Infrastruktur, sowie Hilfestellungen bei der Installation der entsprechenden Systeme zur Verfügung standen.

Koblenz, im März 2012

Petra Schubert

Integration von CRM-Systemen mit Kollaborations-Systemen am Beispiel von DocHouse und Lotus Quickr

Projektpraktikumsdokumentation

Projektpraktikum WS2010-2011
Arbeitsgruppe Betriebliche Anwendungssysteme
Institut für Wirtschafts- und Verwaltungsinformatik
Universität Koblenz-Landau

Von:

Björn Lilge	blilge@uni-koblenz.de
Ludwig Paulsen	lpaulsen@uni-koblenz.de
Marco Wolf	marcowolf@uni-koblenz.de
Markus Aldenhövel	maldenhoevel@uni-koblenz.de
Martin Surrey	msurrey@uni-koblenz.de
Mike Reuther	mreuther@uni-koblenz.de
Roland Diehl	rdiehl@uni-koblenz.de

Inhaltsverzeichnis

1	Motivation.....	5
2	Definition und Funktionsübersicht	6
2.1	CRM-System	6
2.2	Kollaborations-System.....	6
3	Bedarfsanalyse	7
3.1	Verkaufstrichter.....	7
3.2	Workflow Continuum	10
3.3	Kollaborationsschema	10
4	Beschaffenheitsanalyse.....	12
4.1	Synchronisation	12
4.2	Alternative	13
5	Szenario.....	13
5.1	Geschäftssicht	13
5.2	Anwendungssicht	14
5.3	Techniksicht.....	14
5.4	Anwendungsszenario	15
5.4.1	Mögliche Szenarien	15
5.4.2	Gewähltes Szenario	16
6	Synthese.....	16
6.1	Projekt	17
6.2	Person.....	18
6.3	Dokument.....	19
6.4	Aktivität	20
7	Umsetzung	21
7.1	Schnittstellenevaluation.....	21
7.1.1	Lotus QuickPlace Java API	22
7.1.2	Lotus Quickr REST API.....	23
7.1.3	Lotus Quickr Web Services	24
7.1.4	Java/CORBA Klassen	26
7.2	Anforderungsspezifikation	28
7.2.1	Funktionale Anforderungen	29
7.2.2	Nicht-funktionale Anforderungen	30
7.3	Prototypentwicklung.....	31
7.3.1	Beschreibung	31
7.3.2	Implementierungsentwurf	32
7.3.3	Benutzungsoberfläche.....	34
7.3.4	Evaluierung.....	34
8	Fazit.....	36

Abbildungsverzeichnis

Abbildung 1: CRM-System Funktionsübersicht	6
Abbildung 2: Kollaborations-System Funktionsübersicht	7
Abbildung 3: Verkaufstrichter und CRM-Prozess	8
Abbildung 4: Workflow-Continuum ⁵	10
Abbildung 5: Kollaborationsschema I.....	11
Abbildung 6: Kollaborationsschema II.....	11
Abbildung 7: Kollaborationsschema III.....	12
Abbildung 8: Kollaborationsschema IV.....	12
Abbildung 9: Geschäftssicht	13
Abbildung 10: Anwendungssicht.....	14
Abbildung 11: Techniksicht	15
Abbildung 12: Szenarien.....	16
Abbildung 13: Dokumentablagestruktur.....	19
Abbildung 14: Schnittstellenübersicht	21
Abbildung 15: Web Service Übersicht.....	25
Abbildung 16: Klassendiagramm Web Service	26
Abbildung 17: Screenshot Prototyp	31
Abbildung 18: Synchronisationsablauf.....	32
Abbildung 19: Klassendiagramm Protoyp	33

Tabellenverzeichnis

Tabelle 1: Funktionsabgleich DocHouse - Quickr	17
Tabelle 2: Attributabgleich Dokumente	19
Tabelle 3: Attributabgleich Aktivitäten	20
Tabelle 4: REST API Funktionen.....	24
Tabelle 5: Funktionsbeschreibung Buttons.....	32

1 Motivation

In der alltäglichen Praxis existieren viele kollaborative Projekte zwischen Unternehmen. Zur Unterstützung dieser Vorhaben, können über die weit verbreiteten CRM-Systeme hinaus auch Kollaborations-Systeme verwendet werden. Beispiele für solche Systeme sind DocHouse und Lotus Quickr. Um die Zusammenarbeit so effektiv wie möglich zu gestalten, ist das Ziel dieser Arbeit eine Integration zwischen CRM- und Kollaborations-Systemen zu untersuchen und prototypisch zu entwickeln.

Zunächst werden CRM- und Kollaborations-Systeme definiert und eine Funktionsübersicht gegeben. Anschließend wird im Kapitel der Bedarfsanalyse untersucht, inwieweit eine Kollaborationsunterstützung im CRM-Prozess sinnvoll ist. In der Beschaffenheitsanalyse wird die Art der angestrebten Integration diskutiert. Auf Basis dieser Erkenntnisse wird im nächsten Abschnitt ein Szenario entwickelt. Nachfolgend wird die genaue Ausgestaltung der Integration auf Funktionsebene synthetisiert. In der Umsetzung werden die zur Verfügung stehenden Schnittstellen genauer evaluiert. Auf Grundlage des Funktionsabgleiches werden die Anforderungen spezifiziert und der Prototyp vorgestellt und bewertet. Schließlich wird ein Fazit gezogen.

2 Definition und Funktionsübersicht

2.1 CRM-System

Software zum Kundenbeziehungsmanagement (Customer-Relationship-Management), einschließlich Unterstützung von Kundengewinnung und Kundenpflege, wird nach dem Ansatz von Sozio-Technischen-Systemen zusammen mit der zugrunde liegenden Hardware und den Nutzern als CRM-System bezeichnet.

Häufig wird CRM dabei in die Typen Operatives, Analytisches, Kommunikatives und Kollaboratives CRM untergliedert. Wie in Abbildung 1 zu sehen ist, lassen sich die Funktionen eines CRM-Systems ihrem Zweck nach in fünf Kategorien segmentieren. Diese sind wiederum den verschiedenen CRM-Typen zugeordnet. Die mit Stern gekennzeichneten Funktionen sind die primär von dem gewählten Beispiel CRM-System DocHouse gebotenen Funktionen.

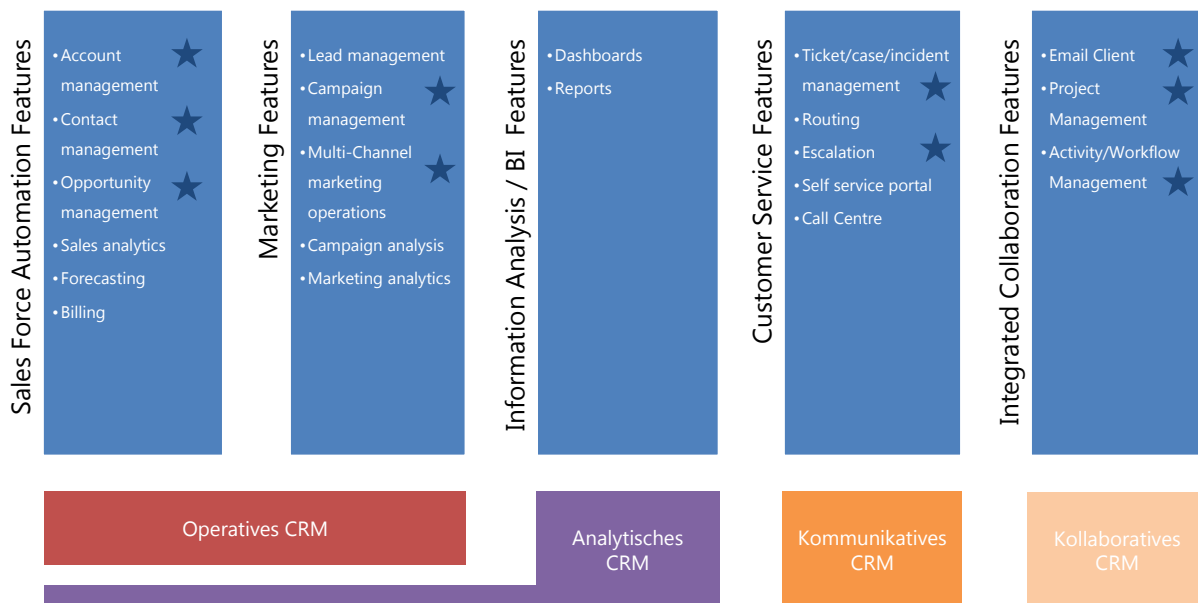


Abbildung 1: CRM-System Funktionsübersicht

2.2 Kollaborations-System

Kollaborations-Systeme, auch Groupware genannt, bezeichnen Informations- und Kommunikationssysteme, die zielgerichtete, aber auch beiläufige Gruppenarbeit unterstützen oder ermöglichen¹. Diese Systeme beinhalten drei Komponenten: Kommunikation, Koordination und Kollaboration.

Da Koordination und Kollaboration stark interdependent sind und Funktionen sowohl koordinativ als auch kollaborativ verwendet werden können, wurde in Abbildung 2 eine Kategorisierung in Kommunikation und Koordination/Kollaboration vorgenommen. Die aufgezeigten Funktionen sind in Anlehnung an eine Marktstudie² ausgewählt. Die mit Stern gekennzeichneten Funktionen sind die primär von dem gewählten Beispiel-Kollaborations-System Lotus Quickr gebotenen Funktionen. Die folgende Betrachtung beschränkt sich auf die Unterstützung durch kollaborative und koordinative Funktionen.

¹ Subsumption von http://www-stud.uni-essen.de/~sw0136/AWi_Seminar.html#Begriff%20Groupware

² Riemer, K. (2007): The Market for E-Collaboration Systems - Identification of Systems Classes using Cluster Analysis, in: 15th European Conference on Information Systems, St.Gallen (CH), 07-09 June 2007

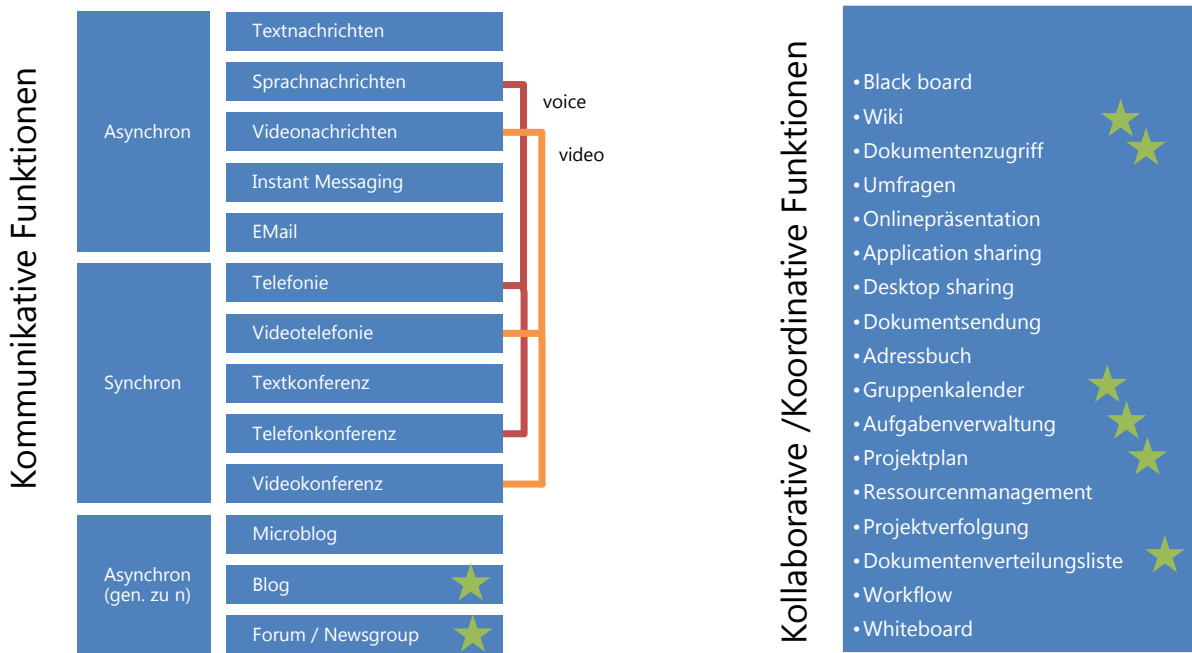


Abbildung 2: Kollaborations-System Funktionsübersicht

3 Bedarfsanalyse

Um ein mögliches Optimierungspotenzial durch eine integrierte Lösung zwischen CRM- und Kollaboration-System zu identifizieren, werden folgend typische CRM-Prozesse untersucht. Auf Grundlage des Verkaufstrichter-Modells erfolgt eine inhaltliche Untersuchung. Des Weiteren werden anhand des Workflow-Continuum-Modells die CRM-Prozesse nach ihrer Beschaffenheit untersucht. Anschließend wird ein allgemeines Kollaborationsschema beschrieben, welches das Integrationspotenzial anhand der Reduzierung von Interaktionsströmen aufzeigt.

3.1 Verkaufstrichter

Im Folgenden wird ein generalisierter Vertriebsprozess (siehe Abbildung 3) schrittweise auf die mögliche Unterstützung durch Kollaborations-Systeme untersucht. Die spezifischen Schritte variieren von Unternehmen zu Unternehmen, aber auch unternehmensintern. Sie enthalten generell die aufgeführten Aspekte. Das gewählte Vertriebsprozessmodell basiert auf der Annahme, dass mit jedem Prozessschritt die Teilmenge der Zielgruppe abnimmt und wird deshalb auch „Verkaufstrichter“ genannt.

Wenn in einer frühen Phase ein Kollaborations-System eingeführt wurde, kann diese natürlich auch in nachfolgenden Phasen verwendet werden.

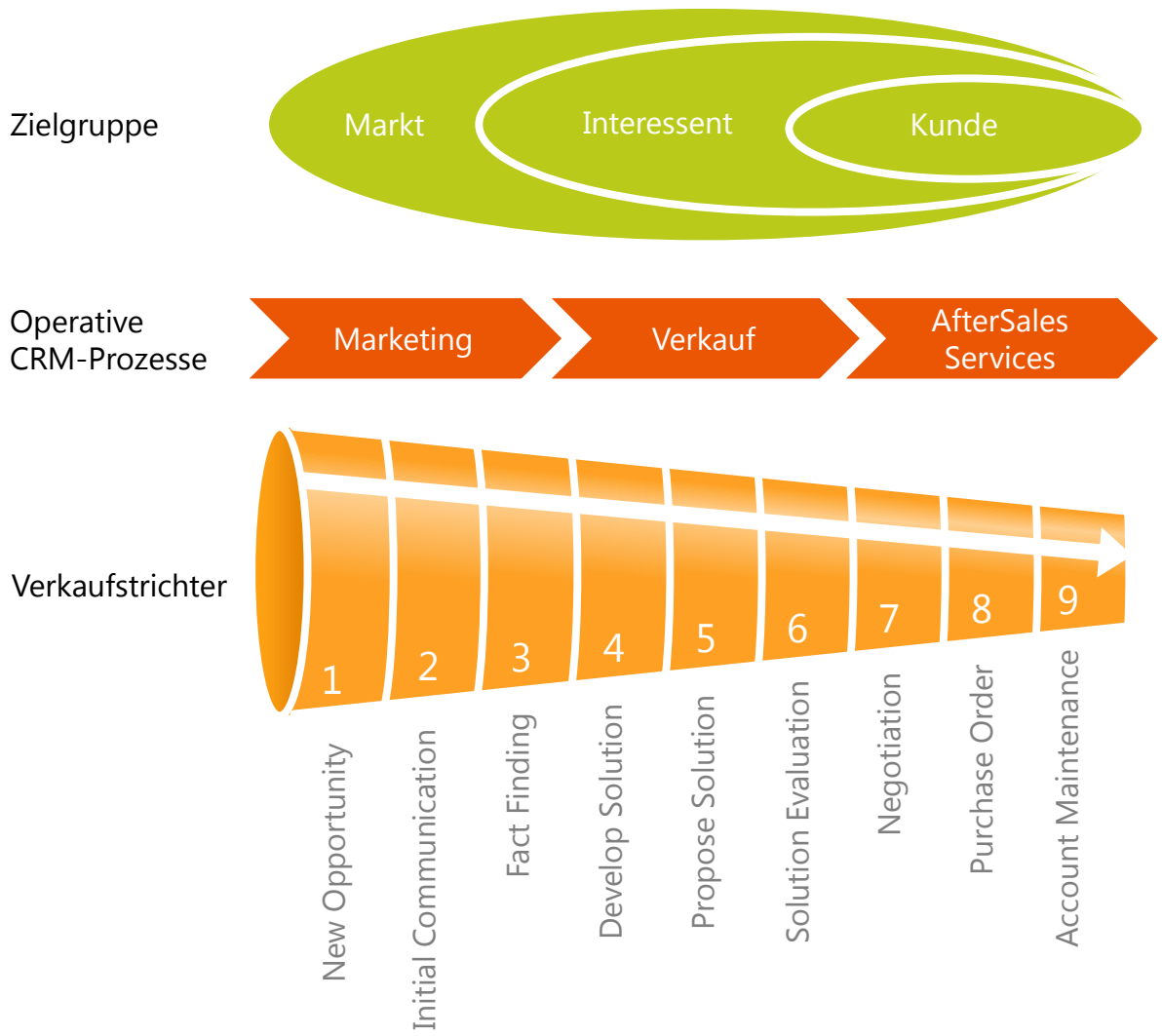


Abbildung 3: Verkaufstrichter und CRM-Prozess³

³ Siehe Meyer, M. (Hrsg.) (2002): CRM-Systeme mit EAI – Konzeption, Implementierung und Evaluation. Wiesbaden: Vieweg. S. 78

<p>1 „New Opportunity“</p>	<p>In diesem ersten Schritt geht es um die Entdeckung neuer potenzieller Kunden. In diesem Prozessschritt findet die Arbeit in der Regel intern oder in einem externen Marktforschungsunternehmen statt. Hier erfolgt also keine Kommunikation mit dem Kunden, weshalb das Kollaborations-System höchstens intern benutzt wird.</p>
<p>2 „Initial Communication“</p>	<p>Der nächste Schritt ist die initiale Kontaktaufnahme mit den zuvor entdeckten, potenziellen Kunden. Da diese Kontaktaufnahme erst der zweite Schritt im Vertriebsprozessmodell ist, fällt die verbliebene Teilmenge der Zielgruppe hier noch sehr groß aus. Einfache kostengünstige Kommunikationswege sind deshalb zu bevorzugen. Die Nutzung eines Kollaborations-Systems ist deshalb für die initiale Kontaktaufnahme, aufgrund einer geringen Rücklaufquote, in der Regel nicht geeignet.</p>
<p>3 „Fact Finding“</p>	<p>Wenn der vorhergehende Schritt mit beidseitiger Interessensbekundung abgeschlossen wurde, findet nun Recherchearbeit über Anforderungen, Potenzial, Risiken und weitere Aspekte des möglichen Verkaufs statt. Ist der Arbeitsumfang in diesem Schritt hoch und externe Interaktion notwendig, kann schon in diesem Schritt ein Kollaborations-System eine Unterstützung darstellen. Zum Beispiel die gemeinsame, versionierte Dokumentenablage oder ein Diskussionsforum. Auch die gemeinsame Erstellung eines Projektplans kann entsprechend unterstützt werden.</p>
<p>4 „Develop Solution“</p>	<p>Diese Stufe ist die Entwicklung des Produktes. Wenn der Kunde in die Entwicklung mit einbezogen wird, gibt es hohen Kommunikations- und Interaktionsbedarf, sodass ein Kollaborations-System unterstützend wirkt.</p>
<p>5 „Propose Solution“</p>	<p>In dieser Phase wird das erstellte Produkt dem Kunden vorgestellt. Kollaborationssoftware kann unterstützend verwendet werden.</p>
<p>6 „Solution Evaluation“</p>	<p>Die Evaluation der Lösung wird durch den Kunden vorgenommen. Ein Kollaborations-System kann zur Aufnahme der Ergebnisse oder zu der Klärung von Fragen dienen. Da die Schritte „Develop Solution“, „Propose Solution“ und „Solution Evaluation“ mehrfach hintereinander ausgeführt werden können, kann ein Kollaborations-System die Dokumentation und den gesamten Verkaufsprozess unterstützen.</p>
<p>7 „Negotiation“</p>	<p>Die Verhandlung über den Abschluss des Verkaufs hat hohe Anforderungen an die Interaktion bezüglich sozialer Aspekte, weshalb Media-Richness sinnvoll sein kann. Basisverhandlungen könnten über ein Diskussionsforum geführt werden. Videokonferenzen bieten hingegen mehr Social Awareness, ein Face-To-Face Meeting wird allerdings einer Verhandlung am ehesten gerecht.</p>
<p>8 „Purchase Order“</p>	<p>Die endgültige Bestellung verläuft standardisiert und wird durch ERP-Systemunterstützung abgewickelt. Ein Kollaborations-System ist hier nicht weiter nützlich.</p>
<p>9 „Account Maintenance“</p>	<p>Falls zu diesem Zeitpunkt ein Kollaborations-System eingesetzt wurde, ist z.B. eine Weiternutzung für Service Fälle möglich.</p>

3.2 Workflow Continuum

Mohan⁴ unterteilt, wie in Abbildung 4 dargestellt, verschiedene Arten von Prozessen nach dem Grad ihrer Strukturiertheit bzw. Flexibilität in ein Klassifikationsschema: das „Workflow-Continuum“.

Ein stark strukturierter Prozess, z.B. ein einfacher Verkaufsvorgang, ist als ein Standardprozess in einem CRM-System implementiert. Kooperative oder ad hoc Prozesse hingegen erfordern individuelle und oder kreative Lösungen, die erst entwickelt werden müssen. Um diesen Entwicklungsprozess bestmöglich zu unterstützen, ist eine Integration von CRM- und Kollaborations-System sinnvoll.

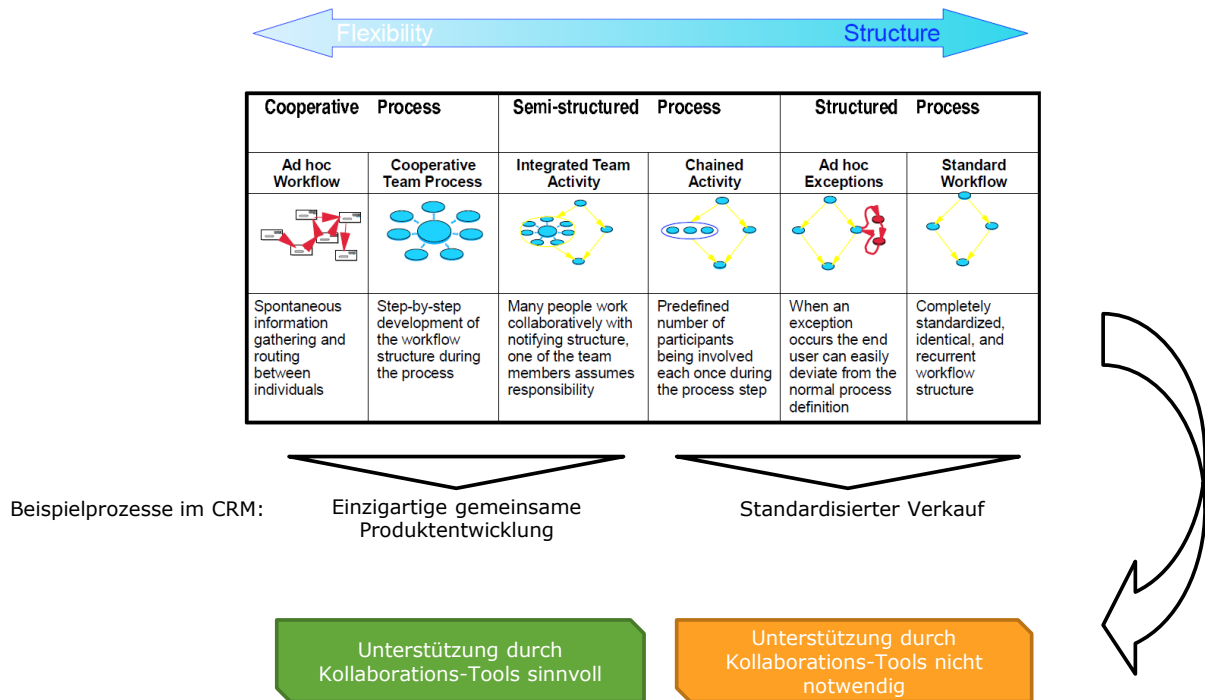


Abbildung 4: Workflow-Continuum⁴

3.3 Kollaborationsschema

In diesem Abschnitt wird Kollaboration im Ablauf des CRM-Prozesses genauer betrachtet. In Abbildung 5 ist dazu Kollaboration auf Instanz-Ebene am Beispiel eines Vertragsschlusses dargestellt. Daraus wird Projektarbeit im Allgemeinen mit ihren verschiedenen Interaktionsströmen (Dokumente, Termine und Organisatorisches sowie inhaltliche Diskussion) schematisch abgeleitet und visualisiert.

⁴ Workflow-Continuum Grafik von Mohan, C. (1999): Workflow Management in the Internet Age

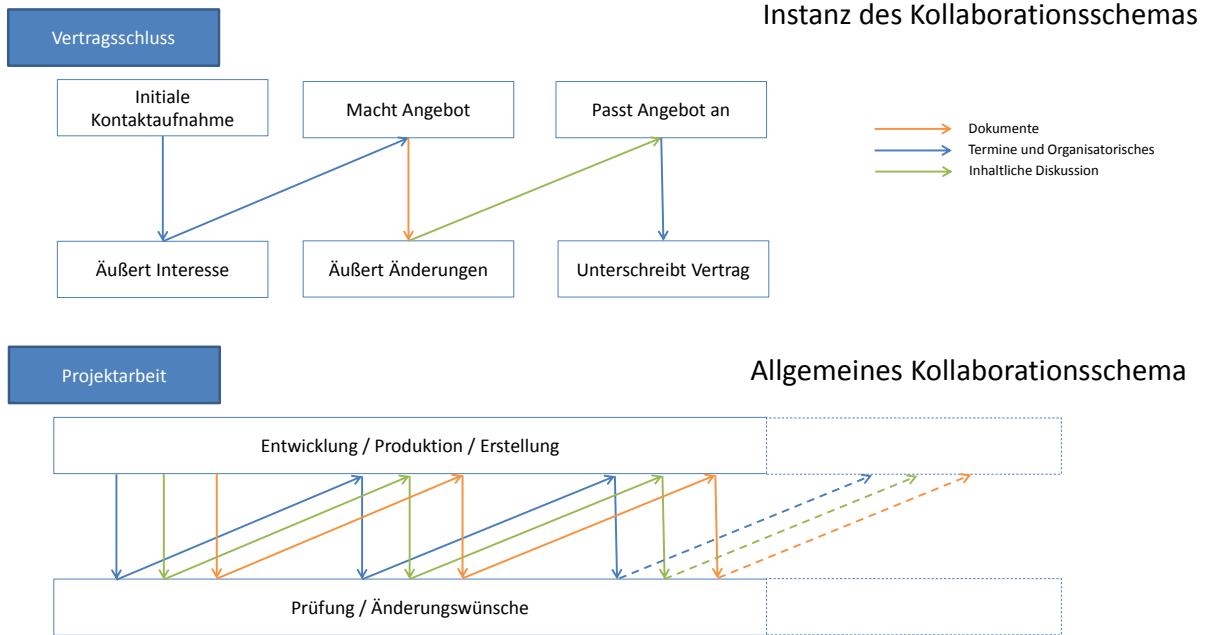


Abbildung 5: Kollaborationsschema I

In Abbildung 6 wird der Zustand bei Nutzung ein Kollaborations-System gezeigt. Statt wie in Abbildung 5 die Interaktionsströme direkt zwischen den Partnern laufen zu lassen, münden diese hier in dem Kollaborations-System. In Abbildung 7 wird das Schema um das intern verwendete CRM-System erweitert. Hier wird deutlich, dass die Ströme redundant verlaufen müssen, um im CRM-System und im Kollaborations-System denselben Informationsstand zu erzielen. Dieses Problem wird in Abbildung 8 gelöst, indem die Integration von CRM-System mit Kollaborations-System dargestellt wird. Somit entstehen keine inhaltlich redundanten Interaktionsströme.

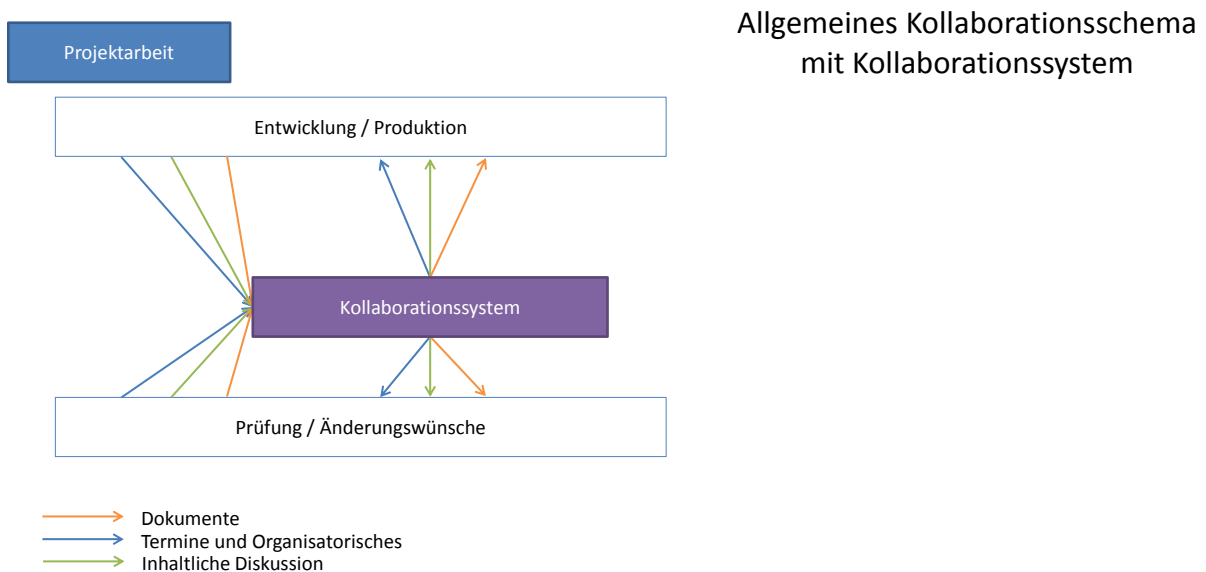
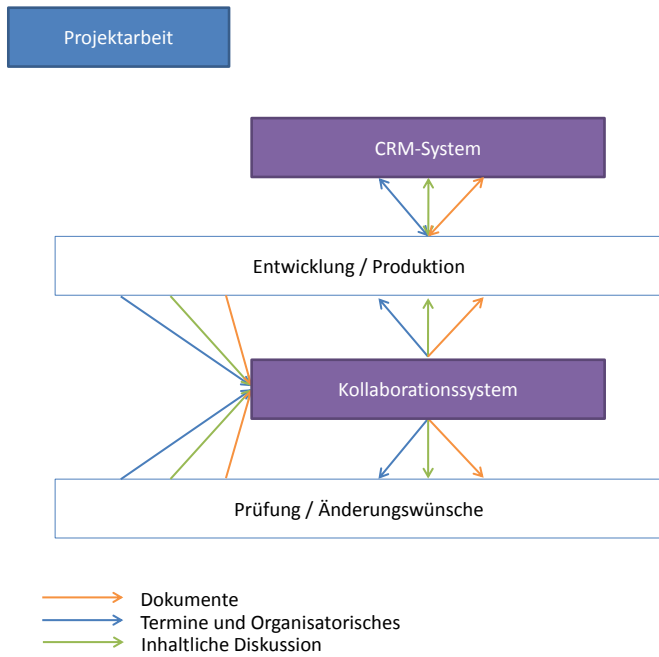
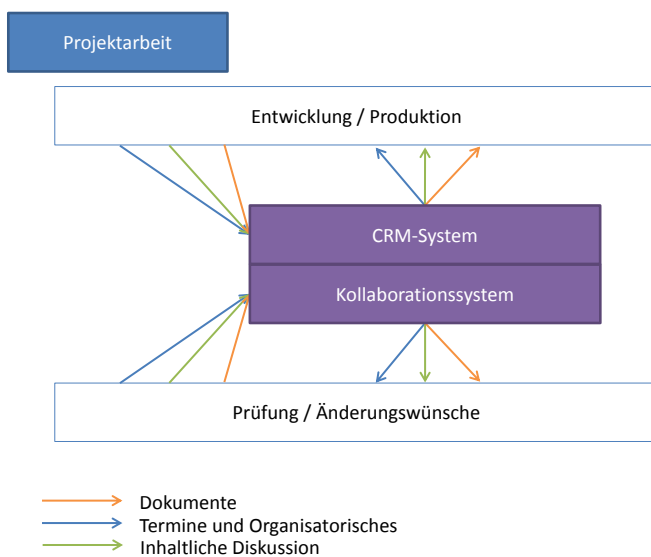


Abbildung 6: Kollaborationsschema II



Allgemeines Kollaborationsschema mit Kollaborationssystem und CRM-System bei einem Partner

Abbildung 7: Kollaborationsschema III



Allgemeines Kollaborationsschema mit Kollaborationssystem und Integration des CRM-System bei einem Partner

Abbildung 8: Kollaborationsschema IV

4 Beschaffenheitsanalyse

Im vorherigen Kapitel wurde aufgezeigt, dass eine Integration zweckmäßig ist. Daraufhin wird nun untersucht, wie diese Integration beschaffen sein sollte.

4.1 Synchronisation

Auf Basis der bisherigen Analyse kann folgendes Integrationsziel aufgestellt werden: Der Nutzer, der sowohl das Kollaborations-System (Quickr) als auch das CRM-System (DocHouse) nutzt, soll zwar beide Systeme simultan verwenden können, jedoch nur mit einem System primär arbeiten.

4.2 Alternative

Neben der Konzeption einer Synchronisierung zwischen den beiden Systemen ist auch die Einführung einer zusätzlichen Ebene denkbar, die Informationen aus DocHouse und Quickr bündelt und in Form eines „Activity Streams“ oder „Social Graphs“ darstellt. Bei diesem Ansatz werden die Risiken der redundanten Datenhaltung bzw. der Synchronisation vermieden. „Activity Streams“ können gefiltert werden, sodass eine geeignete Übersicht erstellt werden kann.

5 Szenario

Im Folgenden wird das dem Projekt zugrunde liegende Szenario detailliert betrachtet und die bei der Erstellung getroffenen Design-Entscheidungen erklärt. Diese Untersuchung wird jeweils aus den drei Perspektiven⁵ Geschäftssicht, Anwendungssicht und Techniksicht vorgenommen.

5.1 Geschäftssicht

Da laut Mohans Workflow Continuum bei kollaborativen Projekten vor allem neuartige Prozesse entstehen, kann keine detaillierte Betrachtung anhand eines Standardprozesses vorgenommen werden. Die Geschäftssicht beschränkt sich deshalb auf die Betrachtung der Makroebene:

Sowohl bei den verschiedenen externen Partnern als auch im eigenen Unternehmen, beginnt der Geschäftsprozess mit der Projektanbahnung. Dieser Schritt umfasst alle Arten von Vorverhandlungen bis hin zum Beginn der Kollaboration.

Die Projektarbeit, die in Abbildung 9 als ein monolithischer Block dargestellt ist, umfasst die komplette Kollaboration und Interaktion aller Geschäftspartner. Sie mündet idealerweise in einem erfolgreichen Projektabschluss, der in diesem Modell auch Teil der Projektarbeit ist.

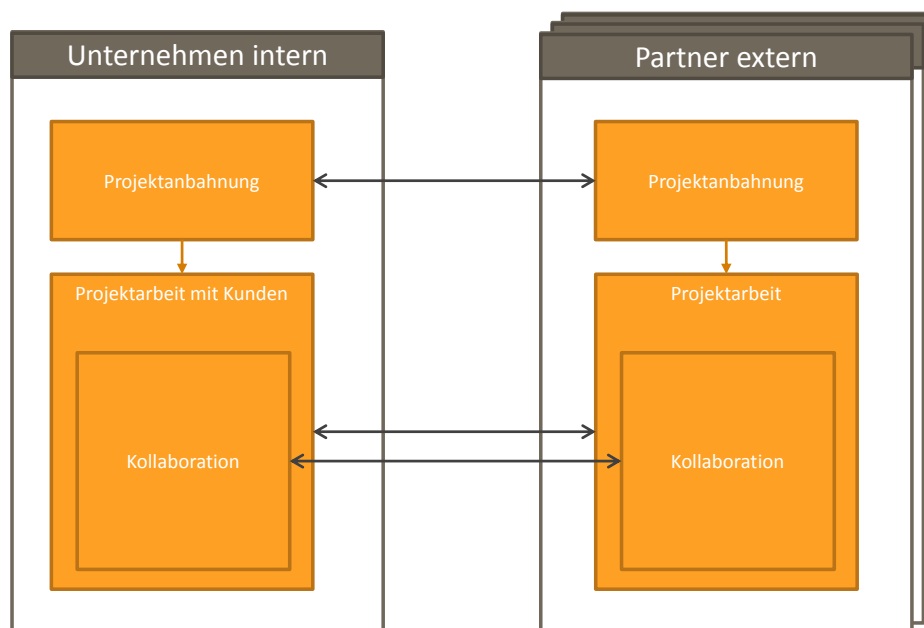


Abbildung 9: Geschäftssicht

⁵ In Anlehnung an: http://www.experience-online.ch/cases/experience20.nsf/de/experience_methodik#4

5.2 Anwendungssicht

In der Anwendungssicht wird die Softwarelandschaft des gewählten Szenarios beschrieben. Die Design-Entscheidung DocHouse und Lotus Quickr auf getrennten Domino Servern bereitzustellen, wurde wegen der größeren Flexibilität in Bezug auf konkrete Business-Szenarien gewählt. Der erste Domino Server stellt die bekannten CRM Funktionen, das Domino Directory 1 und die DocHouse Adressdatenbank bereit. Während der zweite Domino Server die kollaborativen Funktionen, wie Teampplace-Erstellung und Dateiablage, ermöglicht. Befinden sich die beiden Server nicht unter einer Domäne, so können sogar zwei Domino Directories parallel bestehen, wie es in Abbildung 10 dargestellt ist.

Der Zugriff auf das System erfolgt intern über den Browser oder den Lotus Notes Client. Die externen Partner können über ihren Browser auf die entsprechenden Teampplaces zugreifen.

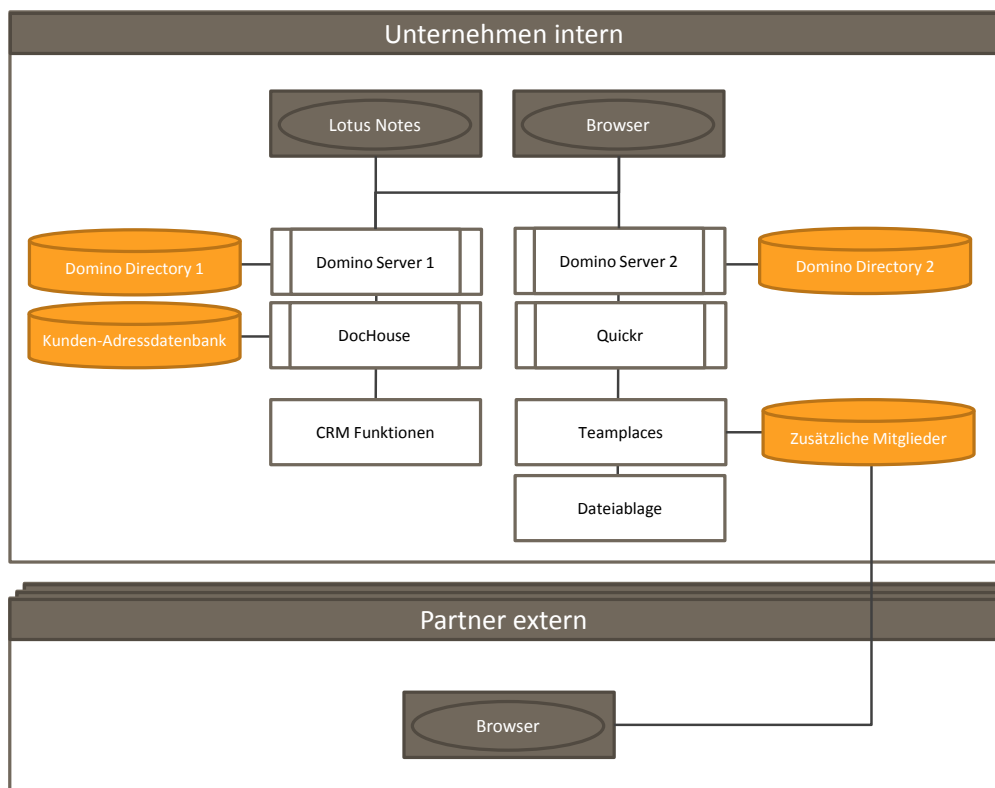


Abbildung 10: Anwendungssicht

5.3 Techniksicht

Die Techniksicht stellt die physische Infrastruktur, die dem gewählten Szenario zugrunde liegt, dar. Externe Partner greifen über das Internet auf Quickr zu. Der Server, der Quickr bereitstellt, liegt in der sogenannten demilitarisierten Zone (DMZ). Ein externer Zugriff auf das von der DMZ losgelöste Intranet ist nicht möglich.

Sowohl Intranet als auch DMZ sind durch eine Firewall nach außen geschützt. Durch dieses Szenario muss die Kommunikation zwischen den beiden Domino Servern nicht extra abgesichert werden. Nur externe Zugriffe durch den Partner müssten durch Aktivierung der SSL Verschlüsselung von Quickr abgesichert werden.

Bei dem Projekt werden folgende Versionen der verschiedenen Software eingesetzt:

- Microsoft Windows Server 2008 R2 64 Bit
- IBM Lotus Domino Server 8.5.1 Fix Pack 5
- IBM Lotus Quickr for Domino 8.5
- DocHouse Version 3.4

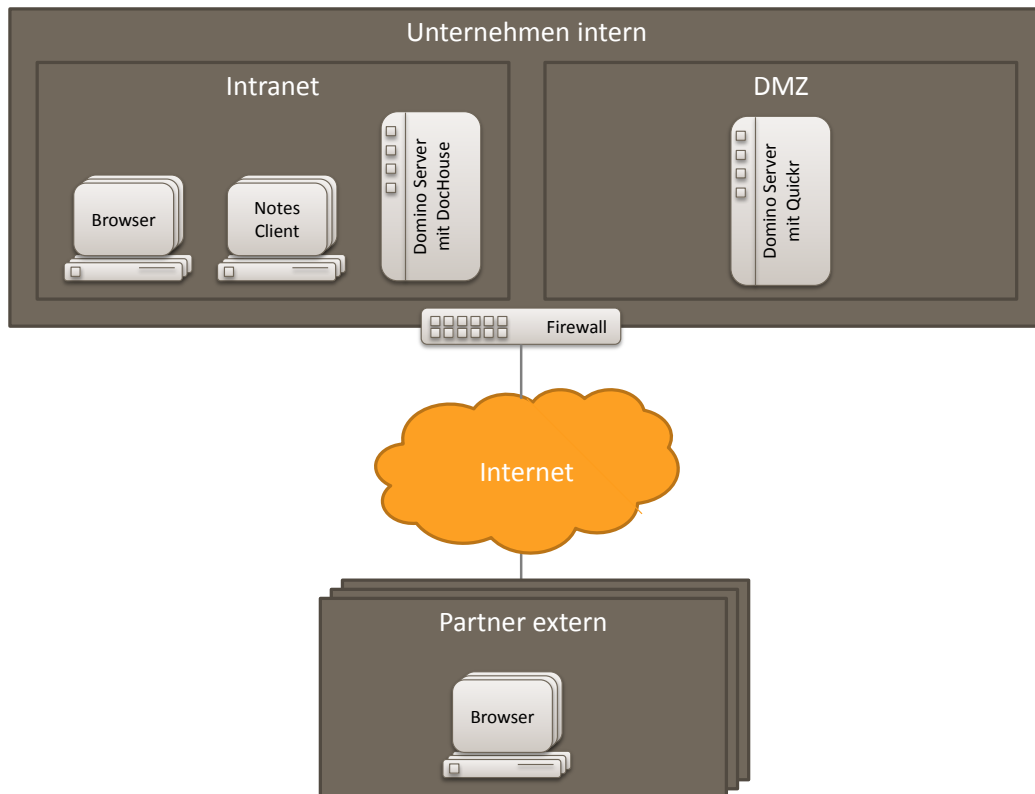


Abbildung 11: Techniksicht

5.4 Anwendungsszenario

5.4.1 Mögliche Szenarien

Folgend werden am Beispiel von Quickr und DocHouse mögliche Integrationsszenarien aufgezeigt:

Eine Zusammenarbeit über Quickr beginnt mit der Einrichtung eines gemeinsamen Teamplaces. Die Daten der teilnehmenden Personen liegen in DocHouse vor und werden als Basis für den Teamplace genutzt.

Wenn nun der Kunde ein Dokument (oder analog einen Termin oder einen Diskussionsbeitrag) erzeugt, geschieht dies alternativlos über Quickr (siehe dazu 2a in Abbildung 12). Daraufhin ist es für eine erfolgreiche Integration nötig, dieses Dokument entweder über redundante Speicherung oder über Verweise in DocHouse einzubinden.

Wenn nun jedoch ein interner Mitarbeiter ein Dokument (o.Ä.) erzeugt, ergeben sich drei Alternativen. Zum einen kann die Anlage direkt in Quickr geschehen (Fall 2a), oder alternativ über DocHouse. Diese Interaktion über DocHouse kann erstens, die Erzeugung in DocHouse mit der Synchronisierung

nach Quickr bedeuten oder zweitens, die Erzeugung in Quickr über die Benutzungsoberfläche von DocHouse (womit dann wieder die in 2a aufgezählten Möglichkeiten: Synchronisation oder Verweise möglich sind).

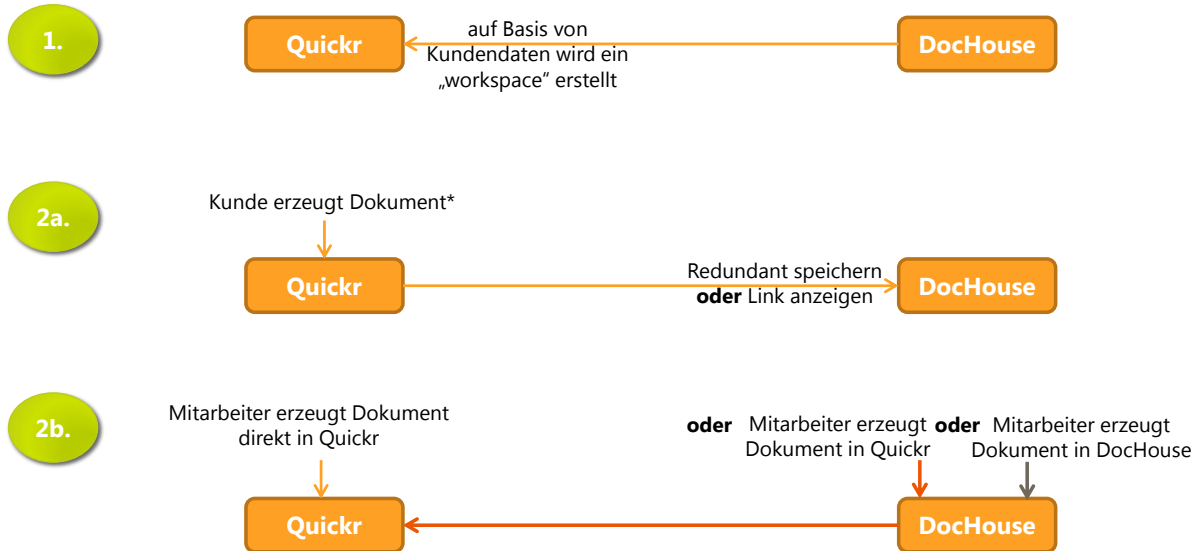


Abbildung 12: Szenarien

5.4.2 Gewähltes Szenario

Ein interner Mitarbeiter erstellt ein Projekt in DocHouse. Diesem Projekt fügt er alle beteiligten internen und externen Projektteilnehmer zu. Der Mitarbeiter erstellt einen Referenzordner, in dem mit Quickr zu synchronisierende Dokumente abgelegt werden. Er ruft die Maske zum Erstellen des Teamplaces auf. Daraufhin werden sowohl der Teamplace als auch die zugehörigen Zugangskonten für die ausgewählten Projektteilnehmer erstellt. Die zu synchronisierenden Dokumente werden anschließend in den Teamplace kopiert. Alle Projektteilnehmer werden per Email über ihren neuen Quickr-Zugang informiert und erhalten somit auch ihre Zugangsdaten.

Der interne Mitarbeiter, der ein Dokument in den Quickr Teamplace stellen möchte, muss beim Hinzufügen über DocHouse zusätzlich zur internen Ablagestruktur nun den Quickr-Referenzordner auswählen.

6 Synthese

Im vorherigen Abschnitt ist zu erkennen, dass Teilschritte des CRM-Prozesses durch ein Kollaborations-System zwischen Unternehmen und Kunden unterstützt werden können. Folgend werden mögliche Integrationsszenarien anhand eines allgemeinen Kollaborationsschemas und auf Basis möglicher Systemarchitekturen analysiert.

Da eine Integration für bestimmte Prozesse sinnvoll ist, gilt es zu überlegen, wie stark die Integration sein sollte bzw. welche allgemeinen Funktionen unterstützt werden sollten. Theoretisch ist eine vollständige Integration anzustreben, da jede Verteilung von Aufgaben, Dokumenten und anderen Ressourcen Mehraufwand bedeutet. Jedoch müssen sowohl die Konzeption als auch die Implementie-

rung von den zu integrierenden Funktionen miteinander kompatibel sein, um eine erfolgreiche Verknüpfung zu gewährleisten.

Im Falle von Lotus Quickr und DocHouse müssen die kompatiblen Funktionen dementsprechend abgeglichen werden.

Tabelle 1: Funktionsabgleich DocHouse - Quickr

DocHouse		Quickr
Projekt	↔	Teampplace
Kontakt	↔	Mitglied
Dokument	↔	Seite
Ordnerstruktur	↔	Bibliotheksstruktur
Aktivität	x	Aufgaben
?	x	Diskussionsforum
?	x	Kalender
...		

Nur eine Teilmenge der möglichen Funktionen kann synchronisiert werden. Bei jedem Abgleich wird die maximale Attributmenge der verschiedenen Klassen synchronisiert. Diese Menge ist immer durch Quickr begrenzt, weil Quickr weniger Attribute als Metadaten speichert. Die unteren Funktionalitäten wie „Aufgaben“, „Diskussionsforum“ und „Kalender“ werden nicht synchronisiert, da sie entweder keine Pendanten haben, keinen Mehrwert liefern oder zu aufwendig in der Implementation sind.

6.1 Projekt

Kollaborative Projekte, wie sie diesem Szenario zugrunde liegen, werden in DocHouse in der Strukturierungseinheit „Projekt“ dargestellt. Die kollaborative Plattform Quickr sieht für die Zusammenarbeit sogenannte „Teampplaces“ vor, welche die einzelnen Funktionalitäten beherbergen. Aus diesem Grund findet die Synchronisierung auf dieser Ebene zwischen DocHouse Projekten und Quickr Teampplaces statt.

Bei der Synchronisierung ist nicht nur auf Zugriffseinschränkungen zu achten, sondern auch zu berücksichtigen, dass eine Übernahme von Projektattributen einen Mehrwert bietet. Absolut ist die Menge der in DocHouse vorhandenen Attribute größer, als die der in Quickr verfügbaren. Bei der Analyse von entsprechenden Pendanten wird die Schnittmenge bei diesem Prototyp auf den Projektnamen und eine Projektbeschreibung reduziert.

Dem DocHouse Projekt zugeordnete Kontakte bilden die Basis für entsprechende Quickr Mitgliederzugänge. Dieser Zusammenhang wird im folgenden Kapitel näher betrachtet.

Nach den Metadaten werden die eigentlichen Arbeitsdaten bzw. Projektelemente, wie Projektstruktur inklusive Aufgaben sowie Projektdokumente, genauer analysiert.

Für den Prototyp wurde ein Teampacetemplate „doctoquickplacetype“ erstellt, welches auf die spezifischen Anforderungen angepasst wurde.

6.2 Person

Personen in Quickr sind sogenannte Mitglieder. Die Mitglieder eines Teamplaces haben verschiedene Rollen (Leser, Autor, Bearbeiter, Manager und Eigentümer), diese bestimmen die Rechte im Teamplace.

Da DocHouse ein CRM-System ist, gibt es unterschiedliche Typen von Personen. Kunden werden in der Regel als Kontakt angelegt. Diese Kontaktdaten zählen zu den Stammdaten in DocHouse. Daher sind Kontakte keine Benutzer im Domino Directory und besitzen damit keine Zugriffsrechte auf Dokumente. Des Weiteren gibt es die Anwender von DocHouse bzw. die Mitarbeiter, die durch das Domino Directory mit Zugriffsrechten ausgestattet sind.

Die externen Kollaborationsteilnehmer werden durch Kontaktdaten aus DocHouse repräsentiert und bekommen Zugriff durch Mitgliedschaft bei dem Quickr Teamplace. Bei der Synchronisation ist es deshalb sinnvoll, die in DocHouse vorhandenen Kontaktdaten eines Projektes als Quickr-Mitglieder zu exportieren. Außerdem sollte zumindest der Projektleiter als Verantwortlicher ebenfalls übertragen werden. Die Kontaktattributmenge von DocHouse übersteigt die Attributmenge aus Quickr deutlich. Aus diesem Grund sollte man sich für die Synchronisation auf die folgende Attribute beschränken:

- Vor- und Nachnamen
- Benutzernamen
- E-Mail-Adresse
- Passwort

Aufgrund der verschiedenen Rollen in Quickr ist es wichtig, dass man bei der Übertragung der Kontakte diese differenziert vergibt. Der Hauptkontakt und der Projektleiter sollten, da sie in der Regel die Verantwortlichen sind, mit Manager Rechten ausgestattet werden. Die weiteren Kontakte sollten mindestens Editor Rechte besitzen. Idealerweise sollten die Projektverantwortlichen das Recht besitzen ein Teamplace zu erstellen. Alternativ könnte man einen Masteruser anlegen, der ausschließlich über dieses Recht verfügt, um eine verschärfte Zugriffskontrolle zu gewährleisten. In DocHouse existiert die Möglichkeit, Personen verschiedenen Projektorganisationseinheiten bzw. Projektteams zuzuweisen. Hier zugeordnete Personen können auch als Grundlage für Mitglieder des Teamplaces dienen. Dabei muss jedoch zwischen Manager, Editor und Reader unterschieden werden. Da Kontakte kein Passwort besitzen, muss für den Quickr Zugriff zunächst eins festgelegt werden. Dieser Vorgang kann auch automatisiert werden. Die Benutzer können ihr Passwort nachträglich ändern.

6.3 Dokument

In DocHouse werden Dokumente einer oder mehreren Projektstrukturen zugewiesen und in einem oder mehreren Ordnern abgelegt. Da komplexe Projektstrukturen in Quickr jedoch schwer abbildbar sind, ist für eine erfolgreiche Integration die Ordnerstruktur von DocHouse zu übernehmen. (siehe Abbildung 13)

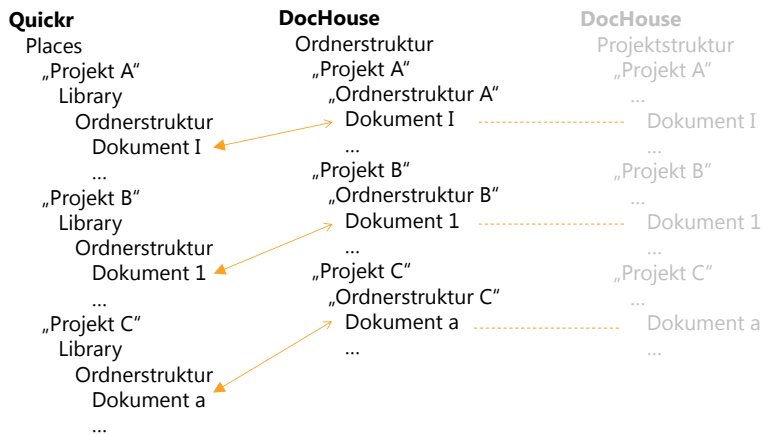


Abbildung 13: Dokumentablageliste

DocHouse-Dokumente können durch sogenannte „Seiten“ in Quickr repräsentiert werden. Folgende Tabelle 2 zeigt einen Abgleich der Eigenschaften von DocHouse-Dokumenten und Quickr Seiten (Quickr Content). In Quickr können diese mit einem Titel erstellt werden, einen Rich-Text als Beschreibung und weitere Dateianhänge erhalten. Seiten haben demnach mit DocHouse-Dokumenten die größtmögliche Attribut-Schnittmenge und stellen deshalb die Entsprechung zu DocHouse-Dokumenten dar.

Tabelle 2: Attributabgleich Dokumente

Quickr Content	DocHouse Dokument
Titel	Titel
Ordner	Ordner (n)
	Kategorie (n)
	Projekte (n)
Ersteller	Eigentümer
letzter Bearbeiter	
	Bearbeiter (n)
	Leser (n)
Erstelldatum	Erstelldatum
Änderungsdatum	Historie (n<=5)
Rich Text	Rich Text (mit Anhangmöglichkeit)
Anhang (n)	
	Aufbewahrungsdatum
	Aufbewahrungsaktion
	Adresse (n)
	weitere Adresszuordnung (n)
	Workflowdetails

6.4 Aktivität

Die Tabelle 3 zeigt die funktionalen Eigenschaften von Aktivitäten in den Systemen Lotus Quickr und DocHouse. Die Aktivitäten aus DocHouse werden hier mit den Aktivitäten aus Quickr verglichen. Ziel dieses Vergleiches ist es in Erfahrung zu bringen, in wie weit es möglich ist, Aktivitäten aus DocHouse in Quickr zu übertragen.

Gemeinsamkeiten (grün), Ähnlichkeiten (gelb) sowie Unterschiede (rot) wurden zu besseren optischen Darstellung farblich gekennzeichnet.

Tabelle 3: Attributabgleich Aktivitäten

Funktionale Eigenschaften Quickr-Task	Funktionale Eigenschaft DH-Task	Zusätzliche funktionale Besonderheiten bei Projekten in DH
Aufgabentitel:<Name>	Aufgabe:<Text>	Aufgabe:<Text>
Beschreibung:<beschreibender Text>	Hinweise:<Text>	Hinweise:<Text>
Status:<abgeschlossen unabgeschlossen>	Status:<offen in Arbeit erledigt>	Status:<offen in Arbeit erledigt>
Anfangsdatum:<Datum>		
Fälligkeitsdatum:<Datum>	fällig bis:<Datum>	fällig bis:<Datum>
Priorität:<hoch mittel niedrig eine wählen>	Priorität:<A B C>	Priorität:<A B C>
erstellt durch:<Benutzer>	Ersteller:<Benutzer>	Ersteller:<Benutzer>
erstellt am:<Datum><Uhrzeit>	Erstelldatum:<Datum><Uhrzeit>	Erstelldatum:<Datum><Uhrzeit>
aktualisiert durch:<Benutzer>	Historie:<letzte Änderungen>	Historie:<letzte Änderungen>
aktualisiert am:<Datum><Uhrzeit>		
im Kalender anzeigen:<1 0>	Erinnerungen:Wiedervorlage am:<Datum>	Erinnerungen:Wiedervorlage am:<Datum>
zugewiesen:<Benutzer>	Verantwortlicher:<Benutzer Benutzergruppe>	Verantwortlicher:<Benutzer Benutzergruppe>
Wer darf bearbeiten:<Benutzer>	Zugriff: Eigentümer:<Benutzer>, Bearbeiter:<Benutzer>, Leser:<Benutzer>, Bearbeitung erlaubt:<Sekretariat Vertretung>, Vertretung:<Benutzer>, Sekretariat:<Benutzer>	Zugriff: Eigentümer:<Benutzer>, Bearbeiter:<Benutzer>, Leser:<Benutzer>, Bearbeitung erlaubt:<Sekretariat Vertretung>, Vertretung:<Benutzer>, Sekretariat:<Benutzer>
	Als Workflow: Kategorie: <Produktion Projekt Kunden usw. >	
istMeilenstein:<1 0>		
	Prüfungsstatus:<keine Prüfung> <zu prüfen> <geprüft:negativ> <geprüft:positiv>	Projektaktivität:In Projekten können Arbeitspakete erstellt werden, die weitere Aktivitäten bzw. Arbeitspakete enthalten. Im Rahmen eines Projektes kommen weitere Eigenschaften hinzu. Status:<geplant> <offen in Arbeit erledigt>,<Fortschritt:<unbekannt> <plan gemäß> <abweichend> <kritisch>,<Fortschritt in % ca.,Kommentar z. Fortschritt:<Text>,<PLAN:<Datum>,<IST:<Datum>,<Start:<Datum>,<Ende<Datum>,<Kosten:<Text>
	Bearbeitungshinweise Ergebnisse:<Text>	
	Unteraktivitäten (weitere Aktivitäten, die Teil der Aktivität sind)	
	DocID:<Zahl>	

Geschachtelte Aktivitäten bzw. Workpackages, sowie sich wiederholende Aktivitäten, lassen sich in Quickr nicht sinnvoll umsetzen. Aus diesem Grund ist es nicht möglich die Aktivitäten vollständig zu integrieren.

7 Umsetzung

7.1 Schnittstellenevaluation

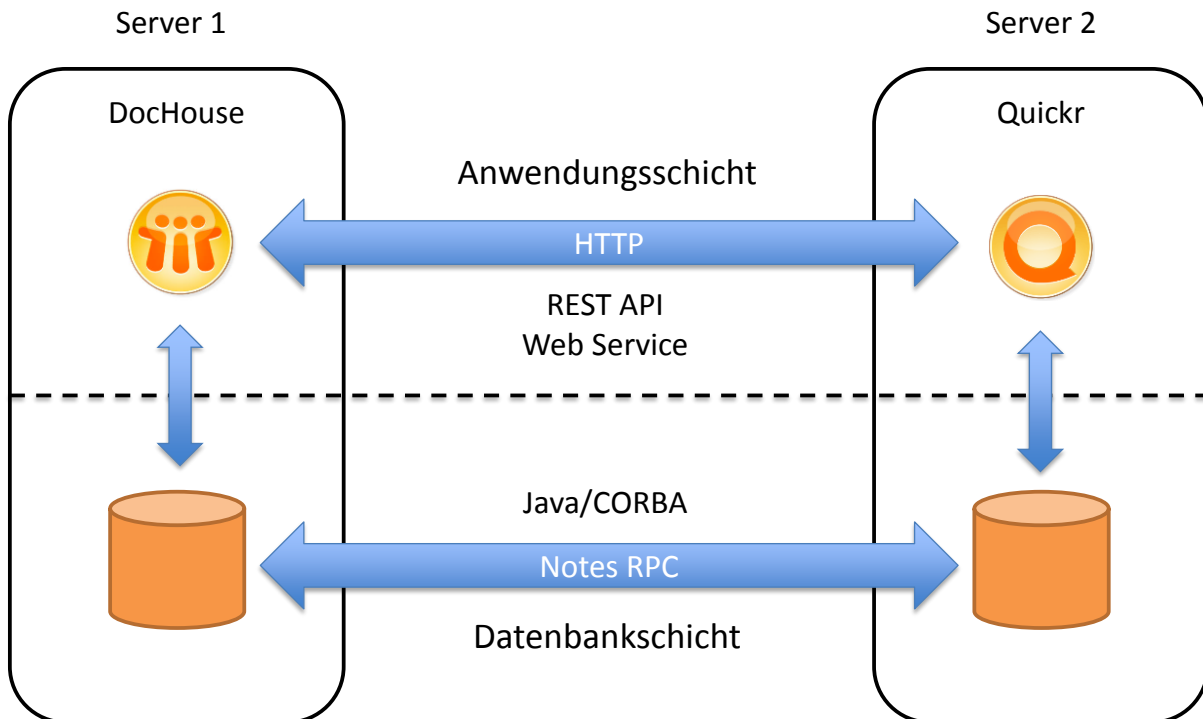


Abbildung 14: Schnittstellenübersicht

Die obere Abbildung beschreibt die verfügbaren Schnittstellen, um Daten zwischen zwei Domino Servern auszutauschen. Es werden nur die Schnittstellen abgebildet, die während der Entwicklung des Prototyps erfolgreich getestet bzw. verwendet wurden. Hierzu gehören die Quickr REST API, der Quickr Web Service und die Java/CORBA Klassen. Die Lotus QuickPlace Java API wird deshalb nicht aufgeführt und in diesem Abschnitt nur beschrieben.

Generell kann man die verwendeten Schnittstellen zwei Schichten zuordnen. Zu der Anwendungsschicht gehören die Quickr REST API und der Web Service, die einen Datenaustausch per HTTP ermöglichen. Auf dieser Schicht findet daher ein Teil der Kommunikation der Anwendungen statt. Bei Lotus Quickr ist das der verwendete Web Browser, über den Teamplaces, Ordner und Dokumente auf der Benutzungsoberfläche bearbeitet werden. Die gleichen HTTP Anforderungen, die der Web Browser an die Quickr Datenbank schickt, können daher auch z.B. per REST API von einem andern Server geschickt werden.

Die Java/CORBA Klassen befinden sich hingegen auf der Datenbankschicht und benutzen zur Übertragen der Daten das Notes RPC Protokoll. Sie können verwendet werden, um im Hintergrund Informationen zwischen Datenbanken von verschiedenen Servern zu übertragen. Im Prototyp wird über diese Schnittstelle die Verbindung zu der DocHouse Projektdatenbank und der Quickr Datenbank hergestellt.

Die genauen Funktionsweisen dieser Schnittstellen und deren Relevanz für die Umsetzung werden im folgenden Abschnitt beschrieben.

7.1.1 Lotus QuickPlace Java API

Die Lotus QuickPlace Java API bietet verschiedene Funktionen wie z.B. Suchoperationen in Teamplaces des Services, Listenerstellung von allen Teamplaces eines bestimmten Servers und die Erstellung von neuen Teamplaces, Benutzern oder Benutzergruppen. Die Verwendung erfolgt über XML.

Um auf die Schnittstelle zugreifen zu können, ist zunächst eine Konfiguration des Betriebssystems notwendig. Der XML Code, welcher die Lotus QuickPlace Objekte spezifiziert, kann danach auf dem Lotus QuickPlace Prozessor ausgeführt werden.

Insgesamt gibt es drei Methoden, um auf die Lotus QuickPlace Java API zuzugreifen.⁶

- Das erstellte XML File kann auf dem Lotus QuickPlace Prozessor von der Kommandozeile oder der Domino Server Konsole ausgeführt werden.
- Die Erstellung des XML Codes kann von einem Java Programm übernommen werden, welches den Code dann auf dem Prozessor ausführt.
- Das QPtool ist ein Server Task, der mit Argumenten in der Kommandozeile oder der Domino Server Konsole aufgerufen werden kann.

Für die Umsetzung wurde die zweite Methode ausgewählt und genauer evaluiert. Hierbei konnten folgende Erkenntnisse gewonnen werden.

Die Klasse „*com.lotus.quickplace.api.QPAPI*“ stellt folgende Methoden zur Verfügung:

- *QPAPI.init()* – Diese Methode wird einmal für den Prozess zu Beginn aufgerufen.
- *QPAPI.process()* – Diese Methode liest das XML Dokument ein und führt es auf dem Lotus QuickPlace Prozessor aus.
- *QPAPI.term()* – Diese Methode terminiert den Vorgang und gibt die Ressourcen wieder frei.

Die Lotus QuickPlace Java API stellt folgende Object APIs zur Verfügung:

- Den service node
- Den server node
- Den place node
- Den placetype node
- Den person node
- Den group node
- Den member node

Alle nodes besitzen verschiedene Attribute und Operationen. Zu den Operationen des place nodes gehören z.B. „create“ und „remove“. Der member node unterstützt die Aktionen „add“ und „remove“. Somit ist die Möglichkeit gegeben, aus einem Java Agenten einen Quickr Teamplace zu erstellen und diesem Mitglieder hinzuzufügen. Die jeweiligen Attribute der nodes definieren Werte wie Titel des Teamplaces oder den Namen eines Mitgliedes.

⁶ http://publib.boulder.ibm.com/infocenter/lqkrhelp/v8r0/index.jsp?topic=/com.ibm.lotus.quickr.admin.dom.doc/dev/qp_dev_api_t.html

7.1.1.1 Relevanz für die Umsetzung

Bei der Umsetzung wurde die Lotus QuickPlace Java API getestet und ein Agent erstellt, der den XML Code auf dem Quickr Prozessor ausführen soll. Hierfür wurde der Beispielcode aus dem Developer's Guide verwendet.⁷ Leider trat bei der Benutzung der QPAPI Klasse eine nicht eindeutige Exception auf. Der Fehler konnte aufgrund von mangelnder Dokumentation und Beispielen nicht gelöst werden. Zusätzlich bedarf es für die Verwendung dieser Schnittstelle eine vorherige Konfiguration des Betriebssystems, damit die Schnittstelle zur Verfügung steht. Die Realisierung, über die Lotus QuickPlace Java API, wurde deshalb als ungeeignet eingestuft.

7.1.2 Lotus Quickr REST API

Lotus Quickr bietet einen REST Service an, der die Erstellung von kollaborativen Lösungen ermöglicht. Der REST Service verwendet die offenen Standards XML und HTTP und macht Ressourcen wie Teamplices, Benutzer oder Dokumente über URLs adressierbar. Diese können daher per HTTP Requests angesprochen werden. Somit ist es beispielsweise möglich mit POST Requests neue Teamplices zu erstellen und diesen Mitglieder hinzuzufügen und Dokumente anzulegen.

Der Quickr REST Service kann mit folgender URL aufgerufen werden:

```
http://<host>:<port>/dm/atom/rest/places/feed
```

Dieser Feed gibt ein XML Dokument zurück, das alle erstellten Teamplices beinhaltet. Für jeden Teamplice Eintrag im Dokument werden auch die REST URLs für Rollen, Mitglieder und Document Libraries ausgegeben.

- Roles: /dm/atom/library/[place_id]/roles
- Members: /dm/atom/library/[place_id]/members
- Document Libraries: /dm/atom/libraries/[place_id]/feed

Beispielsweise können per GET Request an die Members URL alle Benutzer eines Teamplices in XML ausgegeben werden.

⁷ IBM Lotus QuickPlace Developer's Guide, Version 7.0 for Windows, AIX, Solaris and i5/OS, First Edition (August 2005). S. 10

Die folgende Tabelle zeigt weitere Beispiele, wie die REST API für die Erstellung von Ressourcen verwendet werden kann:

Tabelle 4: REST API Funktionen

Methode	URL	Funktion
POST	/dm/atom/rest/places/feed	Teemplate erstellen
POST	/dm/atom/library/[place_id]/action	Mitglied erstellen
POST	/dm/atom/library/[place_id]/feed	Dokument erstellen
POST	/dm/services/DocumentService	Dokument löschen Dokument auschecken Dokument einchecken
POST	/dm/atom/library/[place_id]/document/[document_id]/feed	Neue Version für Dokument erstellen
POST	/dm/atom/library/[place_id]/page/[document_id]/feed	Kommentar für ein Dokument erstellen
POST	/dm/atom/library/[place_id]/page/[document_id]/entry	Version eines Dokumentes wiederherstellen

In dem entsprechenden XML Code können die Werte der jeweiligen Attribute (z.B. Teemplate Namen oder Vorname, Nachname und E-Mail Adresse des Benutzers) definiert werden. Die Verwendung dieser Methoden ist daher dank des XML Standards sehr flexibel und übersichtlich.

7.1.2.1 Relevanz für die Umsetzung

Die REST API ist durch die Anzahl der verfügbaren Operationen die flexibelste aller Schnittstelle, um Daten zwischen DocHouse und Lotus Quickr auszutauschen. Sie bietet viele Möglichkeiten um Ressourcen zu erstellen, zu bearbeiten oder zu löschen. Bei der Umsetzung wurden die Funktionen für die Erstellung der Teemplates, der Mitglieder und der Dokumente über die REST API realisiert. Da die offizielle REST API Dokumentation von IBM an relevanten Stellen unvollständig und mit sehr kurzen Beispielen veranschaulicht ist, fiel die Einarbeitungszeit sehr umfangreich aus. Besonders hilfreich bei der Einarbeitung waren die Entwicklertools des Internet Explorers. Sie wurden verwendet, um den HTTP Verkehr der Quickr Anwendung im Browser aufzuzeichnen. Dabei hat sich herausgestellt, dass auch hier die REST API verwendet wird. Die entsprechenden HTTP Requests konnten somit inklusive der URLs und XML-Anforderungstexten aufgezeichnet und verwendet werden.

7.1.3 Lotus Quickr Web Services

Der Lotus Quickr Web Service stellt insgesamt drei verschiedene Services zur Verfügung, die verschiedene Operationen ermöglichen.

- Der *ContentService* beinhaltet Operationen, um Server Informationen oder Benutzer auszulesen.
- Der *DocumentService* bietet Operationen für die Bearbeitung und Erstellung von Dokumenten und Ordnern an.
- Der *LibraryService* verfügt über Operationen, um Informationen von Libraries des Quickr Servers auszulesen. Mit diesem Service lassen sich beispielsweise alle Teemplates des Servers ausgeben.

Der Web Service verwendet SOAP als Datenaustauschformat zwischen den Anwendungen und ist somit „standardisiert, plattform- und programmiersprachenunabhängig“.⁸

Das WSDL Dokument ist über folgende URL erreichbar⁹:

`http://<host>:<port>/dm/services/ContentService?wsdl`

In dem WSDL Dokument sind alle Definitionen der Operationen und aller komplexen Datentypen des Web Services enthalten. Im Domino Designer lässt sich dieses Dokument auch grafisch und übersichtlich darstellen.

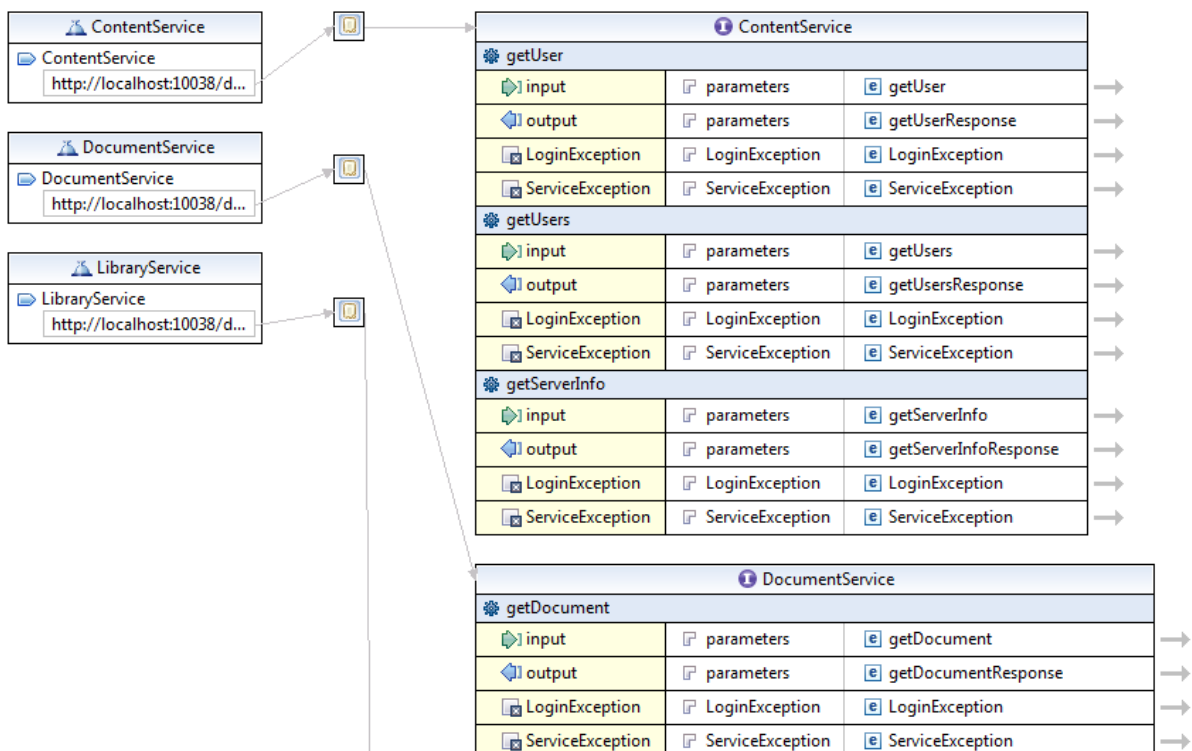


Abbildung 15: Web Service Übersicht

Der Lotus Quickr Web Service kann als Web Service Konsument in einen Java Agenten integriert werden. Der Domino Designer erstellt auf Basis des WSDL Dokumentes die Klassen automatisch. Die jeweiligen angebotenen Funktionen der Services werden in den drei Interfaces ContentService_Port, DocumentService_Port und LibraryService_Port definiert. Zur Verwendung der Services werden die folgenden drei Klassen benötigt, die die oberen Interfaces implementieren:

- ContentServiceSOAPStub
- DocumentServiceSOAPStub
- LibraryServiceSOAPStub

⁸ Laudon, K.C.; Laudon, J.P. und Schoder, D. (2010): Wirtschaftsinformatik – Eine Einführung, 2. Auflage, München: Pearson Studium. S. 264

⁹ <http://www.ibm.com/developerworks/lotus/library/quickr-web-services/>

Jede der aufgerufenen Funktionen liefert ein dazugehöriges „Response“ Objekt zurück. Die „ClbDocumentResponse“ verfügt z.B. über eine Variable des Typs „ClbDocument“. Die Klasse „ClbDocument“ stellt ein Dokument dar, das Attribute wie *URL* oder *documentType* besitzt. Das folgende Klassendiagramm stellt diesen Aufbau grafisch dar.

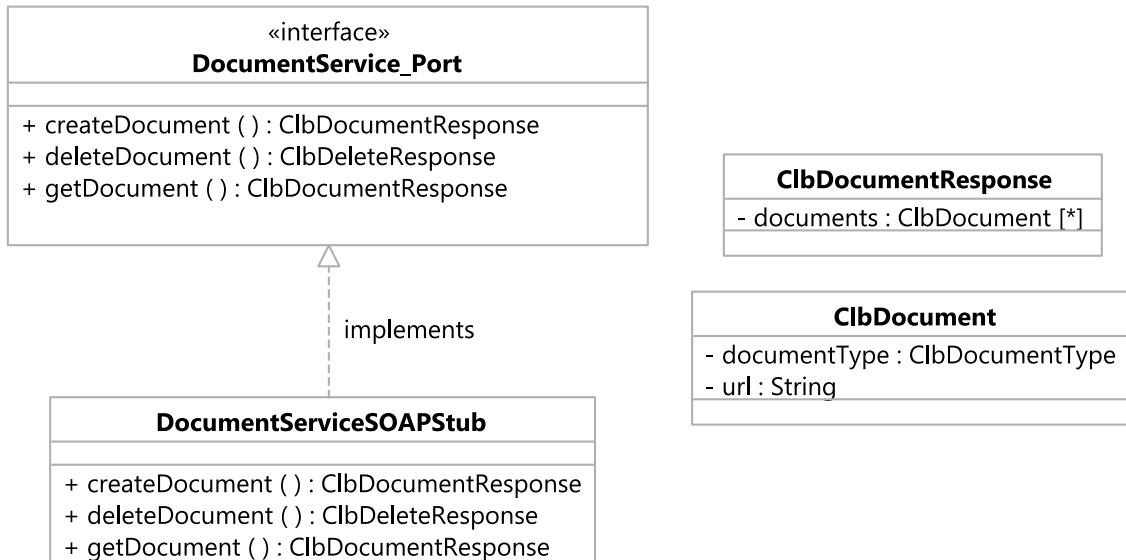


Abbildung 16: Klassendiagramm Web Service

7.1.3.1 Relevanz für die Umsetzung

Der Lotus Quickr Web Service deckt alle erforderlichen Funktionen ab, um auf Libraries und Dokumente zuzugreifen. Die Verwendung des Services ist in der Praxis jedoch komplexer, als z.B. die der REST API. Besonders zu dieser Schnittstelle gibt es auf den offiziellen Seiten von IBM am wenigsten Dokumentation oder Implementierungsbeispiele. Die richtige Anwendung der Klassen und ihrer Funktionen stellte sich daher als schwierig dar, weshalb der Schwerpunkt auf die Verwendung der REST API gelegt wurde.

7.1.4 Java/CORBA Klassen

Die Klassenbibliothek dient zur Entwicklung von entfernten Anwendungen, die auf einen Domino Server zugreifen. Die remote procedure calls werden über das IIOP (Internet Inter-ORB Protocol) erstellt, sodass es möglich ist, diese über TCP/IP Funktionen auf einem Domino Server aufzurufen.¹⁰

Um eine Anwendung mit den Java/CORBA Klassen zu schreiben, muss der Dienst IIOP auf dem zu erreichenden Domino Server laufen. Zum Starten des Services kann dieser über die Domino Konsole angesprochen werden oder durch Anpassung der `notes.ini` aktiviert werden. Dort muss folgendes Attribut angepasst werden:

```
ServerTasks=<andere Tasks>,diiop
```

¹⁰ http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp?topic=/com.ibm.designer.domino.main.doc/H_JAVA_NOTES_CLASSES_JAVA.html

Java/CORBA Klassen werden ebenfalls bei der Programmierung von Agenten in NSF-Dateien verwendet, wenn der Agent in der Programmiersprache Java geschrieben wird. Die Beschreibung begrenzt sich auf die Nutzung der Java/CORBA Klassen in NSF-Dateien.

Die Klasse *Session* bildet die Grundklasse in der Hierarchie, in welche die Verbindung zum Domino Server hergestellt wird. Sie kann in einem Agent über *this.getSession()* abgefragt werden, da ein Agent die Klasse *AgentBase* erweitert, in der die Funktion implementiert ist. Um eine Session auf einen entfernten Domino Server zu erstellen, muss die statische Methode *createSession(hostString, userString, passwordString)* der Klasse *NotesFactory* aufgerufen werden.

Eine Instanz der Klasse *Session* enthält Metadaten über den verbundenen Domino Server und ermöglicht den Zugriff auf andere Domino Objekte. Es kann zum Beispiel eine Datenbank einer NSF-Datei geöffnet werden.

Für das Auslesen von Informationen aus den Domino Datenbanken sind die Klassen *Database*, *DocumentCollection* und *Document* wichtig. Damit die Daten gezielt abgefragt werden können, ist die Verwendung der Volltextsuche von Domino hilfreich.¹¹ Die Abfragesprache ähnelt der SQL-WHERE Bedingung und ist leicht verständlich.

7.1.4.1 Relevanz für die Umsetzung

Die Java/CORBA Klassen ermöglichen unter anderem das Auslesen von Dokumenten aus Datenbanken, die auf einem Domino Server liegen. DocHouse stellt keine offene Schnittstelle zum gezielten Exportieren von Daten bereit, sodass die Möglichkeit mit den Java/CORBA Klassen die relevanten Daten zu beziehen genutzt wurde. Da das Datenmodell in den DocHouse Datenbanken nicht weiter dokumentiert ist, mussten mithilfe von Java/CORBA Klassen die relevanten Daten exportiert und analysiert werden. Dazu wurden spezielle Agenten erstellt, die entsprechende Funktionen zur Verfügung stellen.

¹¹ http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp?topic=/com.ibm.notes.help.doc/DOC/H_TO_USE_SEARCH_OPERATORS_STEPS.html

7.2 Anforderungsspezifikation

Auf der Grundlage der Konzeption und dem erstellten Szenario wurden Anforderungen für Prototypentwicklung erstellt. Die Anforderungen wurden in funktionale und nicht-funktionale Anforderungen untergliedert. Weiter wurde eine Priorisierung in Must-have, Should-have, Nice-to-have vorgenommen, sodass alle Anforderungen mit der Relevanz für die Entwicklung des Prototypens gekennzeichnet wurden.

Zur leichteren Übersicht wurden die Anforderungen in dreizehn Kategorien eingeteilt und wurden mit eindeutigen Nummern versehen.

Der Liste der funktionalen und nicht-funktionalen Anforderungen ist zu entnehmen, welche Anforderungen letztendlich im Prototyp umgesetzt wurden. Dazu ist die Spalte „Status“ mit Werten „erfüllt“ und „nicht geplant“ vorhanden.

7.2.1 Funktionale Anforderungen

Nr.	Name der Anforderung	Beschreibung	Priorität			Status	Aufwand (1-5)
			Must-have	Should-have	Nice-to-have		
FR1	Generelle Anforderungen bezüglich der Funktionalität des Systems						
FR1.1	Teamplace						
FR1.1.1	Teamplaceerstellung	Auf Basis eines Projektes in DocHouse wird ein Teamplace im Quickr erstellt. Dazu wird der Name übernommen.	x			erfüllt	5
FR1.1.2	Import Teamplacedetails	Die "inhaltliche Projektbeschreibung" wird in den Teamplace übernommen.			x	nicht geplant	1
FR1.1.3	Teamplacekonfigurationsmaske	Über eine Maske kann der Benutzer auswählen welche Daten in den Quickr Teamplace exportiert werden.			x	nicht geplant	5
FR1.1.4	Teamplace Logo Export	Ein im DocHouse-Ordner abgelegtes Logo kann mit exportiert und als Teamplace-Logo gesetzt werden.			x	nicht geplant	2
FR1.2	Mitglieder						
FR1.2.1	Mitgliederimport Projektleiter	Der dem Projekt zugeordnete Projektleiter wird dem Teamplace als Manager hinzugefügt.	x			erfüllt	1
FR1.2.2	Mitgliederimport Kunde-Primärkontakt	Der dem Projekt zugeordnete Kunden-Primärkontakt wird dem Teamplace als Manager hinzugefügt.	x			erfüllt	2
FR1.2.3	Mitgliederimport weitere Kundenkontakte	Die dem Projekt zugeordneten weiteren Kundenkontakte werden dem Teamplace als Autoren hinzugefügt.		x		erfüllt	3
FR1.2.4	Mitgliederimport weitere interne Kontakte	Die dem Projekt zugeordneten weiteren internen Kontakte werden dem Teamplace als Autoren hinzugefügt.		x		erfüllt	3
FR1.2.5	Mitgliederimport sonstige Kontakte	Die dem Projekt zugeordneten weiteren Kontakte werden dem Teamplace als Autoren hinzugefügt.			x	erfüllt	3
FR1.2.6	Mitgliederimport Projektorganisation	Die dem Projekt zugeordneten Projektorganisationen werden dem Teamplace entsprechend als Mitglieder hinzugefügt.			x	nicht geplant	5
FR1.2.7	Mitgliederdetails	Zusätzliche Attribute von Kontakten werden aus DocHouse für die Mitglieder übernommen.			x	nicht geplant	2
FR1.3	Dokumente						
FR1.3.1	Dokumentenexport in Quickr-Seiten	Dokumente werden mit allen Anhängen und Text als Quickr-Seite erstellt.	x			erfüllt (w.a.)	3
FR1.3.2	Ablagestrukturexport nach Quickr	Die Ablagestruktur der Dokumente in DocHouse wird in Quickr übernommen.	x			erfüllt	2
FR1.4	Benutzungsoberfläche						
FR1.4.1	Projektmaske	Aus einer Liste der Projekte kann ausgewählt werden, für welche ein Teamplace erstellt werden soll.	x			erfüllt	3
FR1.4.2	Konfigurationsmaske	Die IP Adresse des Quickr Servers und des DocHouse Servers ist konfigurierbar.		x		erfüllt	1
FR1.5	Synchronisation						
FR1.5.1	Synchronisation von Dokumenten von DocHouse nach Quickr	Dokumente werden bei Änderung in DocHouse als neue Version in Quickr gespeichert.	x			erfüllt	4
FR1.5.2	Synchronisation von Mitgliedern von DocHouse nach Quickr	Alle Kontakte aus DocHouse werden mit den Teamplace-Mitgliedern in Quickr synchronisiert.		x		nicht geplant	5
FR1.5.3	Synchronisation von Dokumenten in beide	Dokumente werden bidirektional nach Änderungsdatum synchronisiert.		x		nicht geplant	5

Nr.	Name der Anforderung	Beschreibung	Priorität			Status	Aufwand (1-5)
			Must-have	Should-have	Nice-to-have		
FR2	Datenmanagement						
FR2.1	DocHouse Export-Datenstruktur	Die exportierten Daten sind in einer Datenstruktur verfügbar.		x		erfüllt	5
FR3	Prozessausführung						
FR3.1	Ausführung Export Button	Der Dokumentenexport wird per Knopfdruck gestartet.	x			erfüllt	1
FR3.2	Ausführung Export ereignisgesteuert	Der Dokumentenexport wird automatisiert gestartet.		x		nicht geplant	4
FR3.3	Aktualisierung der Projektmaske	Die Projektdaten in der Projektmaske werden automatisch aktualisiert.	x			erfüllt	3
FR3.4	Passwortversand	Neue Teamplace Mitglieder erhalten eine Mail mit Benutzername und Passwort für den Teamplace.			x	nicht geplant	4
FR5	Funktionale Anforderungen bezüglich Fehlerbehandlung und Ausnahmen						
FR5.1	Fehlerausgabe in Konsole	Die Fehler werden kommentiert in der Domino-Konsole ausgegeben.	x			erfüllt	1
FR6	Funktionale Anforderungen bezüglich Benutzerverwaltung						
FR6.1	Quickr Masteruser	Ein Quickr Masteruser wird benötigt und ist über die Konfigurationsmaske zu konfigurieren.	x			erfüllt	1

7.2.2 Nicht-funktionale Anforderungen

Nr	Name der Anforderun	Beschreibung	Priorität			Status
			Must-have	Should-have	Nice-to-have	
NFR1	Sicherheit					
NFR1.1	Sichere Übertragung	Übertragungen zwischen dem Quickr Server und dem DocHouse Server sind per SSL gesichert			x	nicht geplant
NFR3	Benutzerfreundlichkeit					
NFR3.1	Implizite Gestaltung	Soweit wie möglich werden bestehenden DocHouse Inhalte übernommen und müssen nicht explizit noch einmal angegeben werden	x			erfüllt
NFR3.2	Verwendung der DocHouse Ordnerstruktur	Die Ablage in einem Quickr Teamplace erfolgt über die bestehenden DocHouse Funktionen.	x			erfüllt
NFR8	Übertragbarkeit / Portabilität					
NFR8.1	Eigenständigkeit	Für die Verwendung der Integrationssoftware müssen an Quickr sowie DocHouse keine Anpassungen vorgenommen werden.	x			erfüllt
NFR9	Wiederverwendbarkeit					
NFR9.1	Erweiterbarkeit	Das Design des DocHouse Exports erlaubt eine Erweiterung auf zusätzliche Systeme.			x	erfüllt

7.3 Prototypentwicklung

In diesem Abschnitt wird auf die Entwicklung des Prototyps eingegangen. Dem Abschnitt Funktionale Anforderungen kann entnommen werden, welche Funktionen umgesetzt wurden.

Während der Entwicklung sind UML-Klassendiagramme zur leichteren Durchführung des Projektes und zur Dokumentation erstellt worden.

Im Folgenden wird der Prototyp beschrieben und seine grundlegenden Funktionen erläutert. Der zweite Abschnitt beschreibt das Design der Funktion zur Integration von DocHouse zu Quickr. Zum Schluss wird der entwickelte Prototyp bewertet.

7.3.1 Beschreibung

Der Prototyp ist eine eigenständige Lotus Notes Anwendung, die auf einem Domino Server laufen kann. Durch die Konfigurationsmaske können wichtige Metadaten, wie URLs bzw. IP-Adressen der Domino Server, auf denen DocHouse und/oder Quickr läuft, angegeben werden. Ebenso muss ein Masteruser, der ausreichend Rechte für die Öffnung und Auslesung von Datenbanken und Administration von Quickr besitzt, im Domino-Verzeichnis erstellt und der Anwendung über die Konfigurationsmaske eingespielt werden. Wie schon erwähnt, ist der Prototyp eigenständig, was bedeutet, dass die Programme DocHouse und Quickr in keiner Weise an die Integrationssoftware angepasst werden müssen.

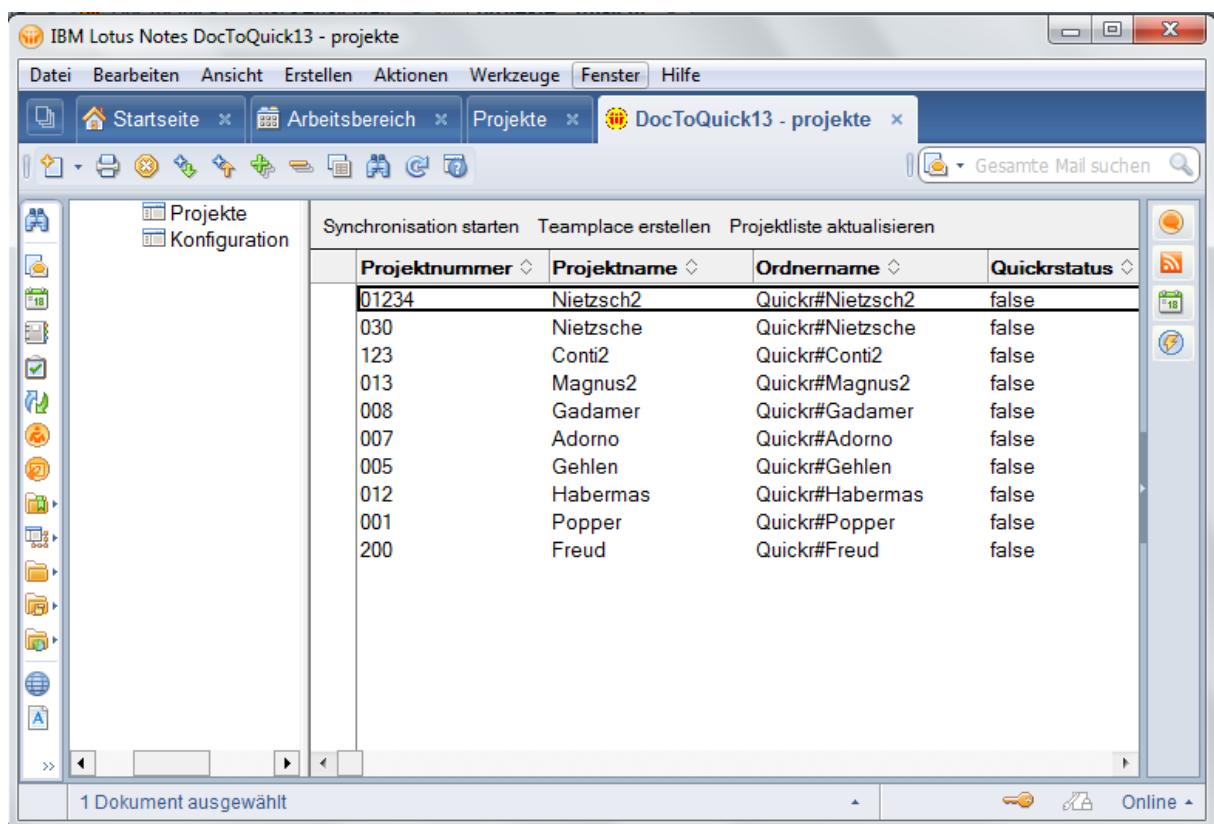


Abbildung 17: Screenshot Prototyp

Die Maske *Projekte* aus Abbildung 17, ist die zentrale Anlaufstelle zum Starten der Integrationsfunktion. Die Anwendung liest alle Projekte aus der DocHouse Datenbank aus und zeigt diese in einer

Liste an. Oberhalb der Liste befinden sich Buttons, deren Funktionen in der folgenden Tabelle kurz erläutert werden.

Tabelle 5: Funktionsbeschreibung Buttons

Bezeichnung	Erläuterung
Synchronisation starten	Selektiert alle Projekte, für die ein Teampplace erstellt werden soll und exportiert alle notwendig Daten, um den zugehörigen Teampplace zu erstellen oder zu aktualisieren.
Teampplace erstellen	Für das ausgewählte Projekt wird bei der nächsten Ausführung von „Integration starten“ ein Teampplace erstellt.
Projektliste aktualisieren	Die Projektliste wird mittels einer Datenbankanfrage an DocHouse auf den neusten Stand gebracht.

Die Abbildung 18 zeigt den generellen Ablauf der Funktion „Synchronisation starten“, die mithilfe des Masterusers ausgeführt wird. Die Daten (Projekte, Dokumente und Adressen) werden aus den DocHouse-Datenbanken exportiert. Dafür werden die Java/CORBA-Klassen der Notes Architektur verwendet. Die exportierten Informationen werden in eine eigens dafür entwickelte Datenstruktur transformiert. Im letzten Arbeitsschritt werden die exportierten Informationen in Quickr gespeichert. Für die Erstellung des Teampplace, das Hinzufügen von Mitgliedern und das Erstellen der Dokumente werden Java/CORBA-Klassen und die REST-API angewendet. Für die einfache Verwendung der REST-API wurde auf Klassen des Apache Abdera Projekts zurückgegriffen.



Abbildung 18: Synchronisationsablauf

7.3.2 Implementierungsentwurf

Die Funktionen, die durch Java Agenten implementiert wurden, sind auf leichte Erweiterbarkeit und Wartbarkeit ausgelegt. Es wurde das Observer-Pattern verwendet, da es sich für die Problemstellung anbietet. Das Observer-Pattern definiert „eine Eins-zu viele-Abhängigkeit zwischen Objekten in der Art, dass alle abhängigen Objekte benachrichtigt werden, wenn sich der Zustand des einen Objekts ändert“¹². Zur Erzielung größtmöglicher Flexibilität wird nicht mit konkreten Klassen, sondern stets auf Schnittstellen programmiert. Die Implementierung dieses Anwendungsfalls nach dem beschriebenen abstrakten Modell, zeichnet sich durch einfache Erweiterbarkeit und Wartbarkeit aus. Der Entwurf ist in Abbildung 19 als UML-Diagramm dargestellt.

¹² Vgl.: Freeman, E.; Freeman, E.; Sierra, K.; Bates, B. (2008): Entwurfsmuster von Kopf bis Fuß. Köln: O'Reilly, S. 51.

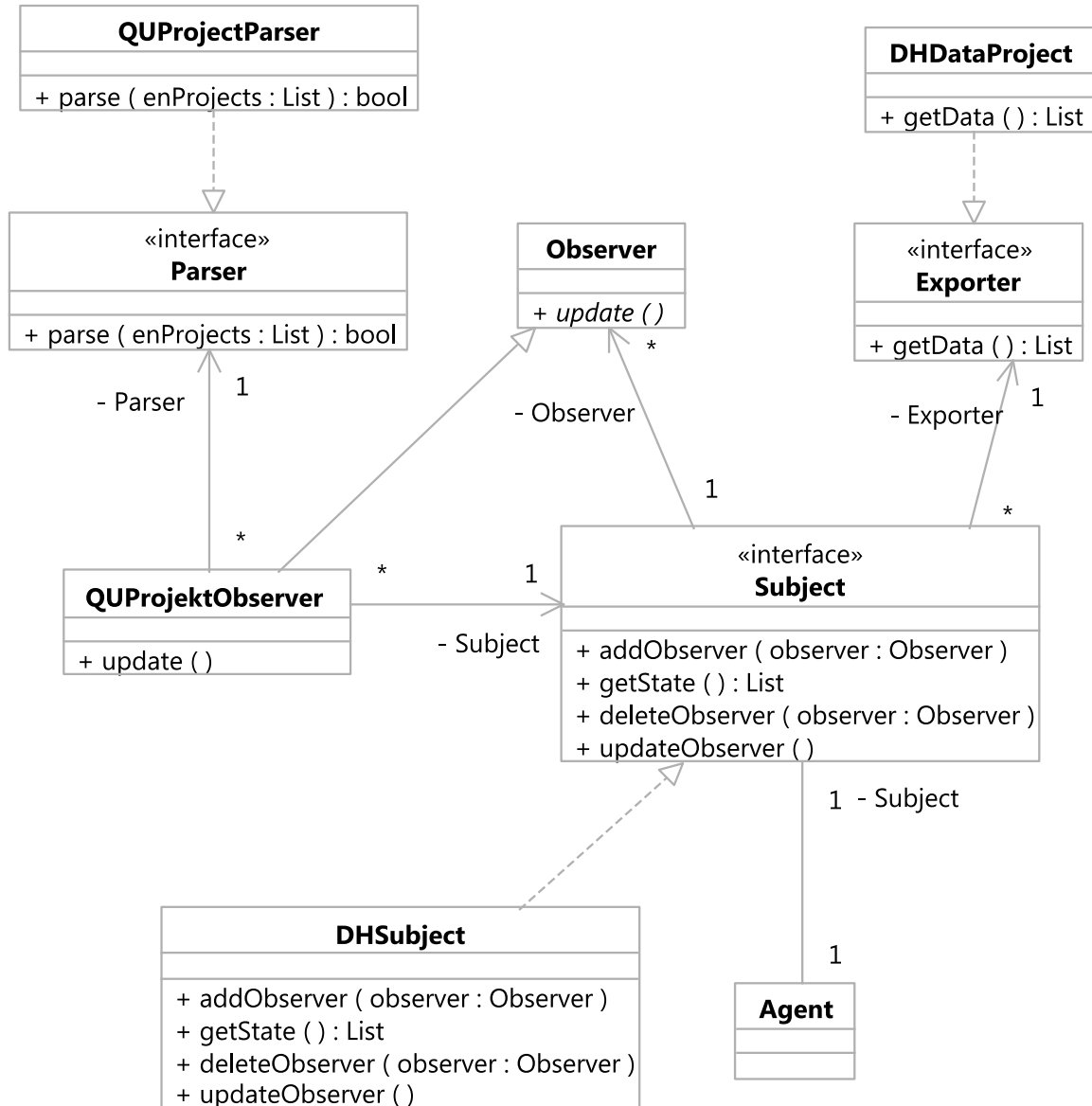


Abbildung 19: Klassendiagramm des Prototypen

Der in Abbildung 18 dargestellte Ablauf wird durch die beiden Akteure DHTSubject und QUObserver implementiert. DHTSubject bietet die Funktionen zum Exportieren von Daten aus DocHouse und zum anschließenden Transformieren. Das Subjekt benachrichtigt alle angemeldeten QUObserver, wenn sich sein Zustand verändert. QUObserver speichert die Informationen in den Quickr. Da sich beliebig viele Observer beim Subjekt anmelden können, ist eine Erweiterbarkeit in Hinblick auf andere Services als Quickr möglich. Das Verhalten der beiden Akteure ist mit den beiden Schnittstellen Exporter und Parser ausgelagert. Diese Auslagerung ermöglicht die zuvor beschriebene leichte Wartbarkeit und Erweiterbarkeit.

7.3.3 Benutzungsoberfläche

Über die Benutzungsoberfläche kann der Anwender auswählen, welche Projekte mit Quickr synchronisiert werden sollen. Diese Auswahl ist essenziell, da nicht jedes Projekt Kollaboration mit dem Kunden erfordert. Eine weitere Oberfläche wird für die Konfiguration der Anwendung durch den Administrator benötigt.

Die Anlage von Dokumenten im Quickr erfolgt explizit in einem dafür vorgesehenen Ordner in der Ablagestruktur von DocHouse. Hierfür ist also keine Benutzungsoberfläche nötig. Über den Prototyp hinaus wäre jedoch denkbar, dass der Benutzer selektieren kann, welche Daten eines DocHouse-Projektes synchronisiert werden sollen, um eventuell sensible Daten ausschließen oder den in Quickr zu erstellenden Mitgliedern im Vorhinein schon Rollen zuweisen zu können.

Notes stellt verschiedene Gestaltungselemente für die Zusammenstellung von grafischen Benutzungsoberflächen zur Verfügung. Die im Prototyp verwendeten Elemente sind Masken, Ansichten, Rahmen und eine Navigationsgliederung.

Masken bieten die Möglichkeit ein Grundgerüst eines Dokuments zu erstellen. Sie sind anpassbar und dienen der späteren Erzeugung und Anzeige von einzelnen Dokumenten. Der Prototyp besitzt eine Maske für die Konfigurationseinstellungen und eine für die Synchronisation von DocHouse-Projekten. Ansichten sind für die Auflistung von Dokumenten verantwortlich. Jede Maske wird im Prototyp durch eine solche Ansicht ergänzt. Die Navigationsgliederung ist für das Umschalten zwischen den einzelnen Ansichten notwendig. Die Rahmen bringen die einzelnen Gestaltungselemente auf einer Oberfläche zusammen und geben eine klare Strukturierung der Oberfläche vor.

Anstatt eine eigene Oberfläche zu gestalten, gäbe es auch die Möglichkeit, die DocHouse-Datenbanken zu verändern und dort an den nötigen Stellen eigene Gestaltungselemente einzuarbeiten. Dies würde jedoch einen erheblichen Aufwand bei der Implementierung der Synchronisationsanwendung nach sich ziehen und bei jeder Versionsänderung von DocHouse eine erneute Durchführung erfordern.

7.3.4 Evaluierung

Der Prototyp setzt die durch das Anforderungsdokument geplanten Funktionen um und bietet darüber hinaus weitere Vorteile. Unter anderem zeigen sich diese bei der Weiterentwicklung des Prototyps. Der softwaretechnische Entwurf und die Verwendung des Observer Musters bieten hohe Flexibilität in Hinblick auf die Erweiterbarkeit und Wartbarkeit. Durch das Entwurfsmuster kann die bidirektionale Synchronisierung leicht in das bestehende Programm eingefügt und verwendet werden. Ebenso ist durch die Auslagerung der Funktionen zum Exportieren und Umwandeln ein hohes Maß an Wartbarkeit gegeben.

Aktuell ist eine implizite Synchronisierung umgesetzt worden, sodass dem Benutzer dieser integrierten Anwendungen die Entscheidung genommen wird, welche Attribute aus DocHouse exportiert werden sollen. Das hat den Vorteil, dass kein Mehraufwand bei der automatischen Teamplaceerstellung entsteht. Dennoch kann es vorkommen, dass diese Entscheidung gezielt getroffen werden muss. Bei den Dokumenten wird eine explizite Synchronisierung angewandt, sodass die DocHouse Benutzer durch den gezielten Speicherort bestimmen, welche Dokumente in der Kollaborationssoftware zur Verfügung stehen sollen.

Mithilfe der Konfigurationsmaske ist es möglich, die Anwendung mit wenigen Anpassungen in einem bestehenden System einzufügen. Zu beachten ist, dass diese Version des Prototyps auf bestimmte Version von DocHouse und Quickr abgestimmt ist.

Aufgrund dessen, dass DocHouse keine offene Schnittstelle, wie einen Webservice oder eine REST-API zur Verfügung stellt, mussten die Daten mithilfe der Java/CORBA Klassen exportiert werden. Sobald Änderungen in DocHouse entstehen, müsste der Prototyp dementsprechend angepasst werden.

Die Integration von DocHouse-Dokumenten mithilfe von Quickr-Seiten ist nicht möglich, da diese Seiten von der REST-API ignoriert werden. Stattdessen werden DocHouse-Dokumente als Ordner angelegt und die entsprechenden Dokumentanhänge als Ordnerinhalt gespeichert.

8 Fazit

Es wurde deutlich, dass durch eine Integration der beiden Systeme die Arbeit mit einem kollaborativen System in den Workflow eines CRM-Systems integriert werden kann und damit redundante Arbeiten vermieden werden. Speziell bietet die automatisierte Generierung von Teamplaces die Grundlage für diese Effizienzsteigerung. Die Synchronisation von Dokumenten in eine gemeinsame Dokumentenablage vertieft diese Integration. Für die Integration weiterer Funktionalitäten von DocHouse und Quickr, wie zum Beispiel Projektstruktur oder Diskussionen, bestehen keine Pendanten. Für eine vollständige Zusammenführung müssten deshalb, über eine Datensynchronisation hinaus, die fehlenden Funktionalitäten erst geschaffen werden.

Bei der Prototypenherstellung stellte sich heraus, dass die reine Verfügbarkeit von Schnittstellen ohne eine hinreichende Dokumentation unzureichend ist. Außerdem ist der Aufwand für die Installation und Konfiguration von Quickr aufgrund von Unzulänglichkeiten der Installationsroutine nicht zu unterschätzen.

Der Prototyp umfasst nur die Synchronisation in eine Richtung. Eine zukünftige Entwicklung könnte eine bidirektionale Verknüpfung mit automatisiertem Abgleich beinhalten.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(<http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte>)

Martin Surrey, Björn Lilge, Ludwig Paulsen, Marco Wolf, Markus Aldenhövel, Mike Reuthel, Roland Diehl, Integration von CRM-Systemen mit Kollaborations-Systemen am Beispiel von DocHouse und Lotus Quickr, Arbeitsberichte aus dem Fachbereich Informatik 3/2012

Martin Surrey, Roland Diehl, DOCHOUSE: Opportunity Management im Partnerkanal (IBM Lotus Quickr), Arbeitsberichte aus dem Fachbereich Informatik 2/2012

Mark Schneider, Ansgar Scherp, Comparing a Grid-based vs. List-based Approach for Faceted Search of Social Media Data on Mobile Devices, Arbeitsberichte aus dem Fachbereich Informatik 1/2012

Petra Schubert, Femi Adisa, Cloud Computing for Standard ERP Systems: Reference Framework and Research Agenda, Arbeitsberichte aus dem Fachbereich Informatik 16/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Natalia A. Zenkova, Simulating social objects with an artificial network using a computer cluster, Arbeitsberichte aus dem Fachbereich Informatik 15/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Simulating medical objects using an artificial network whose structure is based on adaptive resonance theory, Arbeitsberichte aus dem Fachbereich Informatik 14/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, Comparing the efficiency of serial and parallel algorithms for training artificial neural networks using computer clusters, Arbeitsberichte aus dem Fachbereich Informatik, 13/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, A parallel algorithm for selecting activation functions of an artificial network, Arbeitsberichte aus dem Fachbereich Informatik 12/2011

Katharina Bräunlich, Rüdiger Grimm, Andreas Kasten, Sven Vowé, Nico Jahn, Der neue Personalausweis zur Authentifizierung von Wählern bei Onlinewahlen, Arbeitsberichte aus dem Fachbereich Informatik 11/2011

Daniel Eißing, Ansgar Scherp, Steffen Staab, Formal Integration of Individual Knowledge Work and Organizational Knowledge Work with the Core Ontology *strukt*, Arbeitsberichte aus dem Fachbereich Informatik 10/2011

Bernhard Reinert, Martin Schumann, Stefan Müller, Combined Non-Linear Pose Estimation from Points and Lines, Arbeitsberichte aus dem Fachbereich Informatik 9/2011

Tina Walber, Ansgar Scherp, Steffen Staab, Towards the Understanding of Image Semantics by Gaze-based Tag-to-Region Assignments, Arbeitsberichte aus dem Fachbereich Informatik 8/2011

Alexander Kleinen, Ansgar Scherp, Steffen Staab, Mobile Facets – Faceted Search and Exploration of Open Social Media Data on a Touchscreen Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 7/2011

Anna Lantsberg, Klaus G. Troitzsch, Towards A Methodology of Developing Models of E-Service Quality Assessment in Healthcare, Arbeitsberichte aus dem Fachbereich Informatik 6/2011

Ansgar Scherp, Carsten Saathoff, Thomas Franz, Steffen Staab, Designing Core Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 5/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, The prediction of currency exchange rates using artificial neural networks, Arbeitsberichte aus dem Fachbereich Informatik 4/2011

Klaus G. Troitzsch, Anna Lantsberg, Requirements for Health Care Related Websites in Russia: Results from an Analysis of American, British and German Examples, Arbeitsberichte aus dem Fachbereich Informatik 3/2011

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev, A universal simulator based on artificial neural networks for computer clusters, Arbeitsberichte aus dem Fachbereich Informatik 2/2011

Klaus G. Troitzsch, Natalia Zenkova, Alexander Arzamastsev, Development of a technology of designing intelligent information systems for the estimation of social objects, Arbeitsberichte aus dem Fachbereich Informatik 1/2011

Kurt Lautenbach, A Petri Net Approach for Propagating Probabilities and Mass Functions, Arbeitsberichte aus dem Fachbereich Informatik 13/2010

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving“ Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priebe, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereich Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadiou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anforderungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereich Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese, Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extended Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aus dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description: "E-KRHyper", Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Information systems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(<http://www.uni-koblenz.de/fb4/publikationen/gelbereihe>)

Lutz Prieze: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißén: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005